Apriva MESA VPN Server Security Target

14-3117-R-0023

Version: 0.10

7/15/2015

Prepared For:

Apriva ISS, LLC.

8501 North Scottsdale Road, Suite 110

Scottsdale, AZ 85253

Prepared By:

Kenji Yoshino

709 Fiero Lane, Suite 25

San Luis Obispo, CA 93401

Notices:

Table of Contents

Tables

1 Security Target (ST) Introduction

- The ST introduction shall contain an ST reference, a TOE reference, a TOE overview, and a TOE description.
- The ST reference shall uniquely identify the ST.
- The TOE reference shall identify the TOE.

The structure of this document is defined by CC v3.1r3 Part 1 Annex A.2, "Mandatory contents of an ST":

- Section 1 contains the ST Introduction, including the ST reference, Target of Evaluation (TOE) reference, TOE overview, and TOE description.
- Section 2 contains conformance claims to the Common Criteria (CC) version, Protection Profile (PP) and package claims, as well as rationale for these conformance claims.
- Section 3 contains the security problem definition, which includes threats, Organizational Security Policies (OSP), and assumptions that must be countered, enforced, and upheld by the TOE and its operational environment.
- Section 4 contains statements of security objectives for the TOE, and the TOE operational environment as well as rationale for these security objectives.
- Section 5 contains definitions of any extended security requirements claimed in the ST.
- Section 6 contains the security function requirements (SFR), the security assurance requirements (SAR), as well as the rationale for the claimed SFR and SAR.
- Section 7 contains the TOE summary specification, which includes the detailed specification of the IT security functions.

1.1 Security Target Reference

The Security Target reference shall uniquely identify the Security Target.

ST Title:	Apriva MESA VPN Server Security Target
ST Version Number:	Version 0.10
ST Author(s):	Kenji Yoshino
ST Publication Date:	7/15/2015
Keywords:	Network Device, VPN Gateway, IPsec

1.2 Target of Evaluation Reference

The Target of Evaluation reference shall identify the Target of Evaluation.

TOE Developer	Apriva ISS, LLC
	8501 North Scottsdale Road, Suite 110
	Scottsdale, AZ 85253
TOE Name:	Apriva MESA VPN server
TOE Version:	Dell™ PowerEdge™ R720 running Apriva MESA VPN server v1.0 build 21.16

1.3 Target of Evaluation Overview

1.3.1 TOE Product Type

The TOE is classified as a “headend” VPN Gateway Network Device.

1.3.2 TOE Usage

The Apriva MESA VPN server is an IPsec VPN gateway designed to provide mobile devices with a secure connection to a protected network.

The TOE includes the following unevaluated functionality:

- Dell iDRAC7 hardware monitoring and control interface.
- High availability/failover

1.3.3 TOE Major Security Features Summary

- Audit
- Cryptography
- User Data Protection
- Identification and Authentication
- Security Management
- Packet Filtering
- Protection of the TSF
- TOE Access
- Trusted Path/Channels

1.3.4 TOE IT environment hardware/software/firmware requirements

Syslog Server supporting Syslog over TLSv1.2:

- with ciphersuites:
  - TLS_RSA_WITH_AES_128_CBC_SHA
  - TLS_RSA_WITH_AES_256_CBC_SHA
- Conforming to:
  - RFC 5424 (Syslog)
  - RFC 5425 (Syslog over TLS)
  - RFC 5246 (TLSv1.2)

VPN Clients supporting:

- IPsec/IKEv2 (RFC 5996)
  - Authentication with X.509 using:
    - RSA
    - ECDSA
  - Symmetric ciphers:
    - AES-CBC-128
    - AES-CBC-256
  - Integrity Algorithms:
    - HMAC-SHA-256
    - HMAC-SHA-384
    - HMAC-SHA-512

    - Key Agreement
        - Diffie-Hellman Group 14
        - Diffie-Hellman Group 19
        - Diffie-Hellman Group 20
        - Diffie-Hellman Group 24
- IPsec/ESP (RFCs 4301 & 4303)
    - Tunnel Mode
    - Symmetric ciphers:
        - AES-GCM-128
        - AES-GCM-256
    - Integrity:
        - N/A (provided by AES-GCM)

NTP Server

- NTPv4 (RFC 5905)

Local Console:

- RS-232 connection

SSH Client (Remote Console):

- SSHv2 (RFCs 4250, 4251, 4252, & 4253)
- Symmetric Ciphers:
    - AES-CBC-128
    - AES-CBC-256
- Integrity Algorithm:
    - HMAC-SHA-1
- Key Agreement:
    - Diffie-Hellman Group 14 SHA-1
- Server Authentication:
    - SSH_RSA
- Client Authentication:
    - SSH_RSA
    - Password

1.4 Target of Evaluation Description

1.4.1 Target of Evaluation Physical Boundaries

The TOE consists of the following hardware:

- Dell™ PowerEdge™ R720
    - CPU: Intel® Xeon® processor E5-2600 series
    - RAM: 16GB
    - NICs: Qty 4, 1Gb/s
    - Disks: Qty 4, 300GB SAS Hot Pluggable, RAID-1
    - Power Supply: Qty 2, Hot Pluggable
    - CD/DVD: Qty 1, SATA
    - Enhanced Hardware Entropy Generation: QUANTIS PCIe card

Running:

- Apriva MESA VPN server v1.0 build 21.16
  - Red Hat Enterprise Linux 6.5
  - QuickSec
  - OpenSSL FIPS 2.0.5
  - Syslog-ng Premium Edition 5

The guidance documentation that is part of the TOE is listed in Section 9, “References,” within Table 10: TOE Guidance Documentation.

1.4.2 Target of Evaluation Logical Boundaries

The logical boundary of the TOE includes the security functions implemented exclusively by the TOE. These security functions are summarized in Section 1.3.3 above and are further described in the following subsections. A more detailed description of the implementation of these security functions are provided in Section 7, “TOE Summary Specification”.

1.4.2.1 Audit

The TOE generates audit records for security relevant events. The TOE maintains a local audit log as well as sending the audit records to a remote Syslog server. Audit records sent to the remote server are protected by a TLSv1.2 connection. Each audit record includes identity (username, IP address, or process), date and time of the event, type of event, and the outcome of the event. The TOE prevents modification to the local audit log.

1.4.2.2 Cryptographic Operations

The TOE implements CAVP validated cryptographic algorithms for random bit generation, encryption/decryption, authentication, and integrity protection/verification. These algorithms are used to provide security for the SSH, TLS, and IPsec (IKEv2 and ESP) protocols.

The TOE zeroizes all plaintext secret and private cryptographic keys and CSPs once they are no longer required.

1.4.2.3 User Data Protection

The TOE ensures that previous content of network packets is not reused in subsequent network packets. The TOE zeroizes IPsec buffers when the packet has been transmitted. The TOE ensures that all other network buffers are zeroized upon allocation of the buffer.

1.4.2.4 Identification and Authentication

The TOE authenticates administrative users using a username/password combination or a username/SSH_RSA key combination. The TSF does not allow access to any administrative functions prior to successful authentication. The TOE has the capability to lock a remote user’s account if that user exceeds the configured number of failed authentication attempts.

1.4.2.5 Security Management

The TOE implements a limited command line interface (CLI) to allow authorized administrators to configure the TOE. This interface restricts the administrator to executing commands required to configure and administer the TOE.

1.4.2.6 Packet Filtering

The TOE filters packets received on the physical interfaces and virtual interfaces (IPsec tunnels). The TOE reads each packet's header and can be configured to allow or deny the packet based on IPv4 source address, IPv4 destination address, Transport Layer Protocol (if specified in an IPv4 header), TCP or UDP source port, and TCP or UDP destination port.

1.4.2.7 Protection of the TSF

The TOE protects itself through a number of features. The CLI does not provide commands for the administrator to display secret and private keys. The TOE ensures timestamps and timeouts are accurate by maintaining a real-time clock for measuring time as well as polling an NTP server to mitigate drift.

The TOE implements self-tests to verify its correct operation prior to offering protected services (VPN functionality). If the initial self-tests fail or the ongoing health tests fail, the TOE shuts down the VPN functionality and blocks all traffic to or from the network interfaces that were running VPN tunnels.

The TOE automatically verifies the authenticity and integrity of updates by requiring the updates to be digitally signed. TOE verifies that every update is signed by RedHat or Apriva prior to installing the update.

1.4.2.8 TOE Access

The TOE can be configured to display a warning and consent banner when an administrator attempts to establish an interactive session. The TOE also enforces a configurable inactivity timeout for remote administrative and IPsec sessions.

The TOE can be configured to deny establishment of a VPN client session based on the time, day, and/or remote client's IP address.

1.4.2.9 Trusted Path/Channels

The TOE uses IPsec or TLS to provide a trusted communication channel between itself and all authorized IT entities. The trusted channels utilize X.509 certificates to perform mutual authentication. The TOE initiates the TLS trusted channel with the Syslog server while the TOE allows the remote VPN clients to initiate the IPsec trusted channel.

The TOE uses SSH to provide a trusted path between itself and remote administrative users.

1.5 Notation, formatting, and conventions

The notation, formatting, and conventions used in this Security Target are defined below; these styles and clarifying information conventions were developed to aid the reader.

Where necessary, the ST author has added application notes to provide the reader with additional details to aid understanding; they are italicized and usually appear following the element needing clarification. Those notes specific to the TOE are marked "TOE Application Note;" those taken from the ND Protection Profile are marked "PP Application Note."

The notation conventions that refer to iterations, assignments, selections, and refinements made in this Security Target are in reference to SARs and SFRs taken directly from CC Part 2 and Part 3 as well as any SFRs and SARs taken from a Protection Profile.

The notation used in those PP to indicate iterations, assignments, selections, and refinements of SARs and SFRs taken from CC Part 2 and Part 3 is not carried forward into this document. Additionally, obvious errors in the PP are corrected and noted as such.

The CC permits four component operations (assignment, iteration, refinement, and selection) to be performed on requirement components. These operations are defined in Common Criteria, Part 1; paragraph 6.4.1.3.2, "Permitted operations on components" as:

- Iteration: allows a component to be used more than once with varying operations;
- Assignment: allows the specification of parameters;
- Selection: allows the specification of one or more items from a list; and
- Refinement: allows the addition of details.

Iterations are indicated by a number in parenthesis following the requirement number, e.g., FIA_UAU.1.1(1); the iterated requirement titles are similarly indicated, e.g., FIA_UAU.1(1).

Assignments made by the ST author are identified with **bold text.**

Selections are identified with <u>underlined text</u>**.**

Refinements that add text use ***bold and italicized text*** to identify the added text**.** Refinements that performs a deletion, identifies the deleted text with ~~***strikeout, bold, and italicized text***~~.

2 Conformance Claims

2.1 Common Criteria Conformance Claims

This Security Target is conformant to the Common Criteria Version 3.1r3, CC Part 2 extended [2], and CC Part 3 extended [3].

2.2 Conformance to Protection Profiles

This Security Target claims exact compliance to the Protection Profile for Network Devices, Version 1.1, dated June 8, 2012 [6], including the Security Requirements for Network Devices Errata #3, Version 1.0, November 3, 2014 [7]. This Protection Profile and Errata will be referred to as NDPP or PP for convenience throughout this Security Target.

2.3 Conformance to Security Packages

This Security Target extends the NDPP security claims with the Network Device Protection Profile (NDPP) Extended Package VPN Gateway, Version 1.1, dated April 12, 2013 [8]. This Extended Package will be referred to as VPNEP or EP throughout this Security Target. This Security Target is VPNEP-conformant and includes 'headend' requirements from Section 8, Appendix D of the VPNEP.

2.4 Conformance Claims Rationale

To demonstrate that exact conformance is met, this rationale shows all threats are addressed, all OSP are satisfied, no additional assumptions are made, all objectives have been addressed, and all SFRs and SARs have been instantiated.

The following address the completeness of the threats, OSP, and objectives, limitations on the assumptions, and instantiation of the SFRs and SARs:

- Threats
  - All threats defined in the NDPP and EP are carried forward to this ST;
  - No additional threats have been defined in this ST.
- Organizational Security Policies
  - All OSP defined in the NDPP and EP are carried forward to this ST;
  - No additional OSPs have been defined in this ST.
- Assumptions
  - All assumptions defined in the NDPP and EP are carried forward to this ST;
  - No additional assumptions for the operational environment have been defined in this ST.
- Objectives
  - All objectives defined in the NDPP and EP are carried forward to this ST.
- All SFRs and SARs defined in the NDPP and EP are carried forward to this ST.

Rationale presented in the body of this ST shows all assumptions on the operational environment have been upheld, all the OSP are enforced, all defined objectives have been met and these objectives counter the defined threats.

Additionally, all SFRs and SARs defined in the NDPP and EP have been properly instantiated in this ST; therefore, this ST shows exact compliance to the NDPP and EP.

3 Security Problem Definition

3.1 Threats

The following table defines the security threats for the TOE, characterized by a threat agent, an asset, and an adverse action of that threat agent on that asset. These threats are taken directly from the PP and EP unchanged.

Table 1: Threats

Threat	Description
T.ADMIN_ERROR	An administrator may unintentionally install or configure the TOE incorrectly, resulting in ineffective security mechanisms.
T.TSF_FAILURE	Security mechanisms of the TOE may fail, leading to a compromise of the TSF.
T.UNDETECTED_ACTIONS	Malicious remote users or external IT entities may take actions that adversely affect the security of the TOE. These actions may remain undetected and thus their effects cannot be effectively mitigated.
T.UNAUTHORIZED_ACCESS	A user may gain unauthorized access to the TOE data and TOE executable code. A malicious user, process, or external IT entity may masquerade as an authorized entity in order to gain unauthorized access to data or TOE resources. A malicious user, process, or external IT entity may misrepresent itself as the TOE to obtain identification and authentication data.
T.UNAUTHORIZED_UPDATE	A malicious party attempts to supply the end user with an update to the product that may compromise the security features of the TOE.
T.USER_DATA_REUSE	User data may be inadvertently sent to a destination not intended by the original sender.
T.NETWORK_DISCLOSURE	Sensitive information on a protected network might be disclosed resulting from ingress- or egress-based actions.
T.NETWORK_ACCESS	Unauthorized access may be achieved to services on a protected network from outside that network, or alternately services outside a protected network from inside the protected network
T.NETWORK_MISUSE	Access to services made available by a protected network might be used counter to Operational Environment policies.
T.REPLAY_ATTACK	If malicious or external IT entities are able to gain access to the network, they may have the ability to capture information traversing throughout the network and send them on to the intended receiver.
T.DATA_INTEGRITY	A malicious party attempts to change the data being sent – resulting in loss of integrity.
T.UNAUTHORIZED_CONNECTION	While a VPN client may have the necessary credentials (e.g., certificate, pre-shared key) to connect to a VPN gateway, there may be instances where the remote client, or the machine the client is operating on, has been compromised and attempts to make unauthorized connections.
T.HIJACKED_SESSION	There may be an instance where a remote client's session is hijacked due to session activity. This could be accomplished because a user has walked away from the machine that was used to establish the session.
T.UNPROTECTED_TRAFFIC	A remote machine's network traffic may be exposed to a hostile network. A user may be required to use a hostile (or unknown) network to send network traffic without being able to route the traffic appropriately.

3.2 Organizational Security Policies

The following table defines the organizational security policies, which are a set of rules, practices, and procedures imposed by an organization to address its security needs. These threats are taken directly from the PP and EP unchanged.

Table 2: Organizational Security Policies	
OSP	Description
P.ACCESS_BANNER	The TOE shall display an initial banner describing restrictions of use, legal agreements, or any other appropriate information to which users consent by accessing the TOE.

3.3 Assumptions

This section describes the assumptions on the operational environment in which the TOE is intended to be used. It includes information about the physical, personnel, and connectivity aspects of the environment. The operational environment must be managed in accordance with the provided guidance documentation. The following table defines specific conditions that are assumed to exist in an environment where the TOE is deployed. These assumptions are taken directly from the PP and EP unchanged.

Table 3: Assumptions	
Assumption	Description
A.NO_GENERAL_PURPOSE	It is assumed that there are no general-purpose computing capabilities (e.g., compilers or user applications) available to the TOE, other than those services necessary for the operation, administration and support of 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_ADMIN	TOE Administrators are trusted to follow and apply all administrator guidance in a trusted manner.
A.CONNECTIONS	It is assumed that the TOE is connected to distinct networks in a manner that ensures that the TOE security policies will be enforced on all applicable network traffic flowing among the attached networks

4 Security Objectives

4.1 Security Objectives for the TOE

Table 4: Security Objectives for the TOE	
TOE Objective	Description
O.PROTECTED_COMMUNICATIONS	The TOE will provide protected communication channels for administrators, other parts of a distributed TOE, and authorized IT entities.
O.VERIFIABLE_UPDATES	The TOE will provide the capability to help ensure that any updates to the TOE can be verified by the administrator to be unaltered and (optionally) from a trusted source.
O.SYSTEM_MONITORING	The TOE will provide the capability to generate audit data and send those data to an external IT entity.
O.DISPLAY_BANNER	The TOE will display an advisory warning regarding use of the TOE.
O.TOE_ADMINISTRATION	The TOE will provide mechanisms to ensure that only administrators are able to log in and configure the TOE, and provide protections for logged-in administrators.
O.RESIDUAL_INFORMATION_CLEARING	The TOE will ensure that any data contained in a protected resource is not available when the resource is reallocated.
O.SESSION_LOCK	The TOE shall provide mechanisms that mitigate the risk of unattended sessions being hijacked.
O.TSF_SELF_TEST	The TOE will provide the capability to test some subset of its security functionality to ensure it is operating properly.
O.ADDRESS_FILTERING	The TOE will provide the means to filter and log network packets based on source and destination addresses
O.AUTHENTICATION	The TOE will provide a means to authenticate the user to ensure they are communicating with an authorized external IT entity.
O.CRYPTOGRAPHIC_FUNCTIONS	The TOE will provide means to encrypt and decrypt data as a means to maintain confidentiality and allow for detection and modification of TSF data that is transmitted outside of the TOE.
O.FAIL_SECURE	Upon a self-test failure, the TOE will shutdown to ensure data cannot be passed while not adhering to the security policies configured by the administrator.
O.PORT_FILTERING	The TOE will provide the means to filter and log network packets based on source and destination transport layer ports.
O.CLIENT_ESTABLISHMENT_CONSTRAINTS	The administrator may configure the headend VPN gateway to accept a client's request for a connection based on attributes the administrator feels are appropriate.
O.REMOTE_SESSION_TERMINATION	A session termination capability is necessary during an administrator specified time period.
O.ASSIGNED_PRIVATE_ADDRESS	While a user may be connected via an untrusted network, it should still be possible to ensure that it can communicate with a known entity that controls the routing of the client's network packets. This can be accomplished by the VPN headend assigning an IP address that the gateway controls, as well as providing a routing point for the client's network

Table 4: Security Objectives for the TOE	
TOE Objective	Description
	traffic.

4.2 Security Objectives for the Operational Environment

Table 5: Security Objectives for the Operational Environment	
Objective	Description
OE.NO_GENERAL_PURPOSE	There are no general-purpose computing capabilities (e.g., compilers or user applications) available to the TOE, other than those services necessary for the operation, administration and support of the TOE.
OE.PHYSICAL	Physical security, commensurate with the value of the TOE and the data it contains, is provided by the environment.
OE.TRUSTED_ADMIN	TOE Administrators are trusted to follow and apply all administrator guidance in a trusted manner.
OE.CONNECTIONS	TOE administrators will ensure that the TOE is installed in a manner that will allow the TOE to effectively enforce its policies on network traffic flowing among attached networks.

5 Extended Components Definition

This section provides definition of the extended security functional and assurance requirements; the components that are CC Part 2 extended, and CC Part 3 extended, i.e., NIAP interpreted requirements, and extended requirements.

5.1 Extended Security Functional Requirements Definitions

There are no extended Security Functional Requirements defined in this Security Target. All extended SFRs were taken from the PP or EP.

5.2 Extended Security Assurance Requirement Definitions

There are no extended Security Assurance Requirements defined in this Security Target. All extended SARs were taken from the PP.

6 Security Requirements

This section describes the security functional and assurance requirements for the TOE; those that are CC Part 2 conformant, CC Part 2 extended, CC Part 3 conformant, and CC Part 3 extended.

6.1 Security Function Requirements

This section describes the functional requirements for the TOE. The security functional requirement components in this security target are CC Part 2 conformant or CC Part 2 extended as defined in Section 2, Conformance Claims. Operations that were performed in the NDPP and EP are not signified in this section. Operations performed by the ST are denoted according to the formatting conventions in Section 1.5.

Table 6: Security Functional Requirements

#	SFR	Description
1	FAU_GEN.1	Audit Data Generation
2	FAU_GEN.2	User Audit Association
3	FAU_STG_EXT.1	External Audit Trail Storage
4	FCS_CKM.1(1)	Cryptographic Key Generation (Asymmetric Keys)
5	FCS_CKM.1(2)	Cryptographic Key Generation (for asymmetric keys)
6	FCS_CKM_EXT.4	Cryptographic Key Zeroization
7	FCS_COP.1(1)	Cryptographic Operation (Data Encryption/Decryption)
8	FCS_COP.1(2)	Cryptographic Operation (Cryptographic Signature)
9	FCS_COP.1(3)	Cryptographic Operation (Cryptographic Hashing)
10	FCS_COP.1(4)	Cryptographic Operation (Keyed-Hash Message Authentication)
11	FCS_IPSEC_EXT.1	Extended: Internet Protocol Security (IPsec) Communications
12	FCS_TLS_EXT.1	Transport Layer Security
13	FCS_SSH_EXT.1	Secure Shell
14	FCS_RBG_EXT.1	Extended: Cryptographic Operation: Random Bit Generation
15	FDP_RIP.2	Full Resident Information Protection
16	FIA_AFL.1	Authentication Failure Handling
17	FIA_PMG_EXT.1	Password Management
18	FIA_UIA_EXT.1	User Identification and Authentication
19	FIA_UAU_EXT.2	Extended: Password-based Authentication Mechanisms
20	FIA_UAU.7	Protected Authentication Feedback
21	FIA_X509_EXT.1	Extended: X.509 Certificates
22	FMT_MOF.1	Management of Security Functions Behavior
23	FMT_MTD.1	Management of TSF Data (General TSF Data)
24	FMT_SMF.1	Specification of management functions
25	FMT_SMR.2	Security Management Roles
26	FPF_RUL_EXT.1	Packet Filtering

Table 6: Security Functional Requirements		
#	SFR	Description
27	FPT_SKP_EXT.1	Protection of TSF Data (for reading of all symmetric keys)
28	FPT_APW_EXT.1	Protection of Administrator Passwords
29	FPT_FLS.1	Fail Secure
30	FPT_STM.1	Reliable Time Stamp
31	FPT_TUD_EXT.1	Extended: Trusted Update
32	FPT_TST_EXT.1	Extended: TSF Testing
33	FTA_SSL_EXT.1	TSF-initiated session locking
34	FTA_SSL.3	TSF-initiated termination
35	FTA_SSL.3(2)	TSF-initiated Termination
36	FTA_SSL.4	User-initiated termination
37	FTA_TAB.1	Default TOE Access Banners
38	FTA_TSE.1	TOE Session Establishment
39	FTA_VCM_EXT.1	VPN Client Management
40	FTP_ITC.1	Inter-TSF trusted channel
41	FTP_TRP.1	Trusted Path

6.1.1 Security Audit (FAU)

6.1.1.1 FAU_GEN.1 Audit Data Generation

FAU_GEN.1.1 The TSF shall be able to generate an audit record for the following auditable events:

a) Start-up and shut-down of the audit functions;
b) All auditable events for the not specified level of audit;
c) All administrative actions; and
d) Specifically defined auditable events listed in Table 7.

Table 7: Auditable Events			
#	SFR	Auditable Events	Additional Audit Record Contents
1	FAU_GEN.1	None.	
2	FAU_GEN.2	None.	
3	FAU_STG_EXT.1	None.	
4	FCS_CKM.1(1)	None.	
5	FCS_CKM.1(2)	None.	
6	FCS_CKM_EXT.4	None.	
7	FCS_COP.1(1)	None.	
8	FCS_COP.1(2)	None.	
9	FCS_COP.1(3)	None.	
10	FCS_COP.1(4)	None.	

Table 7: Auditable Events			
#	SFR	Auditable Events	Additional Audit Record Contents
11	FCS_IPSEC_EXT.1	Failure to establish an IPsec SA. Establishment/Termination of an IPsec SA.	Reason for failure. Non-TOE endpoint of connection (IP address) for both success and failures.
		Session Establishment with peer[1]	Source and destination addresses Source and destination ports TOE interface
12	FCS_TLS_EXT.1	Failure to establish a TLS Session. Establishment/Termination of a TLS session.	Reason for failure. Non-TOE endpoint of connection (IP address) for both successes and failures.
13	FCS_SSH_EXT.1	Failure to establish an SSH session. Establishment/Termination of an SSH session.	Reason for failure. Non-TOE endpoint of connection (IP address) for both successes and failures.
14	FCS_RBG_EXT.1	None.	
15	FDP_RIP.2	None.	
16	FIA_AFL.1	None.	
17	FIA_PMG_EXT.1	None.	
18	FIA_UIA_EXT.1	All use of the identification and authentication mechanism.	Provided user identity, origin of the attempt (e.g., IP address).
19	FIA_UAU_EXT.2	All use of the authentication mechanism.	Origin of the attempt (e.g., IP address).
20	FIA_UAU.7	None.	
21	FIA_X509_EXT.1	Establishing a session with CA	Source and destination addresses Source and destination ports TOE interface
22	FMT_MOF.1	None.	
23	FMT_MTD.1	None.	
24	FMT_SMF.1	None.	
25	FMT_SMR.2	None.	
26	FPF_RUL_EXT.1	Application of rules configured with the 'log' operation	Source and destination addresses Source and destination ports Transport Layer Protocol TOE interface
		Indication of packets dropped due to too much network traffic	TOE interface that is unable to process packets
27	FPT_SKP_EXT.1	None.	
28	FPT_APW_EXT.1	None.	

[1] EP Application Note: For session establishment, the expectation is that the TOE is capable of auditing all of the packets associated with the establishment of a session; this would include the IKE phase 1 and phase 2 negotiations. The TOE must be able to log all of the packets in a successful session establishment, and also have the ability to log any packets that were dropped or discarded.

Table 7: Auditable Events			
#	SFR	Auditable Events	Additional Audit Record Contents
29	FPT_STM.1	Changes to the time.	The old and new values for the time. Origin of the attempt (e.g., IP address).
30	FPT_TUD_EXT.1	Initiation of update.	No additional information.
31	FPT_TST_EXT.1	None.	
32	FTA_SSL_EXT.1	Any attempts at unlocking of an interactive session.	No additional information.
33	FTA_SSL.3	The termination of a remote session by the session locking mechanism.	No additional information.
34	FTA_SSL.3(2)	None.	
35	FTA_SSL.4	The termination of an interactive session.	No additional information.
36	FTA_TAB.1	None.	
37	FTA_TSE.1	None.	
38	FTA_VCM_EXT.1	None.	
39	FTP_ITC.1	Initiation of the trusted channel. Termination of the trusted channel. Failure of the trusted channel functions.	Identification of the initiator and target of failed trusted channels establishment attempt.
40	FTP_TRP.1	Initiation of the trusted channel. Termination of the trusted channel. Failure of the trusted channel functions.	Identification of the claimed user identity.

PP Application Note: *The ST author can include other auditable events directly in the table; they are not limited to the list presented.*

Many auditable aspects of the SFRs included in this document deal with administrative actions. Item c above requires all administrative actions to be auditable, so no additional specification of the auditability of these actions is specified in Table 7.

Assurance Activity: The evaluator shall check the administrative guide and ensure that it lists all of the auditable events and provides a format for audit records. Each audit record format type must be covered, along with a brief description of each field. The evaluator shall check to make sure that every audit event type mandated by the PP is described and that the description of the fields contains the information required in FAU_GEN.1.2, and the additional information specified in Table 7.

The evaluator shall also make a determination of the administrative actions that are relevant in the context of the NDPP. The evaluator shall examine the administrative guide and make a determination of which administrative commands, including subcommands, scripts, and configuration files, are

related to the configuration (including enabling or disabling) of the mechanisms implemented in the TOE that are necessary to enforce the requirements specified in the PP. The evaluator shall document the methodology or approach taken while determining which actions in the administrative guide are security relevant with respect to the NDPP. The evaluator may perform this activity as part of the activities associated with ensuring the AGD_OPE guidance satisfies the requirements.

The evaluator shall test the TOE's ability to correctly generate audit records by having the TOE generate audit records for the events listed in Table 7 and administrative actions. This should include all instances of an event--for instance, if there are several different I&A mechanisms for a system, the FIA_UIA_EXT.1 events must be generated for each mechanism. The evaluator shall test that audit records are generated for the establishment and termination of a channel for each of the cryptographic protocols contained in the ST. If HTTPS is implemented, the test demonstrating the establishment and termination of a TLS session can be combined with the test for an HTTPS session. For administrative actions, the evaluator shall test that each action determined by the evaluator above to be security relevant in the context of the NDPP is auditable. When verifying the test results, the evaluator shall ensure the audit records generated during testing match the format specified in the administrative guide, and that the fields in each audit record have the proper entries.

Note that the testing here can be accomplished in conjunction with the testing of the security mechanisms directly. For example, testing performed to ensure that the administrative guidance provided is correct verifies that AGD_OPE.1 is satisfied and should address the invocation of the administrative actions that are needed to verify the audit records are generated as expected.

EP Assurance Activity:

TSS: The evaluator shall verify that the TSS describes how the Packet filter firewall rules can be configured to log network traffic associated with applicable rules. Note that this activity should have been addressed with a combination of the TSS assurance activities for FPF_RUL_EXT.1.

The evaluator shall verify that the TSS describes how the TOE behaves when one of its interfaces is overwhelmed by network traffic. It is acceptable for the TOE to drop packets that it cannot process, but under no circumstances is the TOE allowed to pass packets that do not satisfy a rule that allows the permit operation or belong to an allowed established session. It may not always be possible for the TOE to audit dropped packets due to implementation limitations. These limitations and circumstances in which the event of dropped packets is not audited shall be described in the TSS.

Guidance: The evaluator shall verify that the operational guidance describes how to configure the Packet filter firewall rules to result in applicable network traffic logging. Note that this activity should have been addressed with a combination of the guidance assurance activities for FPF_RUL_EXT.1.

Test: The following test is expected to execute outside the context of the other requirements. While testing the TOE's compliance against the SFRs, either specific tests are developed and run in the context of this SFR, or as is typically done, the audit capability is turned on while testing the TOE's behavior in complying to the other SFRs in the EP.

- Test 1: The evaluator shall attempt to flood the TOE with network packets such that the TOE will be unable to process all the packets. This may require the evaluator to configure the TOE to limit the bandwidth the TOE is capable to handling (e.g., use of a 10 MB interface).

FAU_GEN.1.2 The TSF shall record within each audit record at least the following information:

- Date and time of the event, type of event, subject identity, and the outcome (success or failure) of the event; and
- For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, information specified in column three of Table 7.

PP Application Note: *As with the previous component, the ST author should update Table 7 above with any additional information generated. "Subject identity" in the context of this requirement could be either the administrator's user id or the affected network interface, for example.*

Assurance Activity: This activity should be accomplished in conjunction with the testing of FAU_GEN.1.1.

6.1.1.2 FAU_GEN.2 User Identity Association

FAU_GEN.2.1 For audit events resulting from actions of identified users, the TSF shall be able to associate each auditable event with the identity of the user that caused the event.

Assurance Activity: This activity should be accomplished in conjunction with the testing of FAU_GEN.1.1.

6.1.1.3 FAU_STG_EXT.1 External Audit Trail Storage

FAU_STG_EXT.1.1 The TSF shall be able to perform transmit the generated audit data to an external IT entity using a trusted channel implementing the TLS protocol.

PP Application Note: *For applications of the NDPP to TOEs that do not act as audit servers, the TOE relies on a non-TOE audit server for storage and review of audit records. Although the TOE generates audit records, the storage of these audit records and the ability to allow the administrator to review these audit records is provided by the operational environment. The ST author chooses the first clause of the first selection in these cases. The NDPP can also be used to specify requirements for an audit server; in this case, the second clause of the first selection is used.*

In the second selection, the ST author chooses the means by which this connection is protected. The ST author also ensures that the supporting protocol requirement matching the selection is included in the ST.

Assurance Activity: For both types of TOEs (those that act as an audit server and those that send data to an external audit server), there is some amount of local storage. The evaluator shall examine the TSS to ensure it describes the amount of audit data that are stored locally; what happens when the local audit data store is full; and how these records are protected against unauthorized access. The evaluator shall also examine the operational guidance to determine that it describes the relationship between the local audit data and the audit data that are sent to the audit log server (for TOEs that are not acting as an audit log server). For example, when an audit event is generated, is it simultaneously sent to the external server and the local store, or is the local store periodically by sending the data to the audit server.

~~TOE acts as audit server: The evaluator shall examine the TSS to ensure it describes the connection supported from non-TOE entities to send the audit data to the TOE, and how the trusted channel is provided. Testing of the trusted channel mechanism will be performed as specified in the associated assurance activities for the particular trusted channel mechanism. The evaluator shall also examine the operational guidance to ensure it describes how to establish the trusted channel with the TOE, as well as describe any requirements for other IT entities to connect and send audit data to the TOE (particular audit server protocol, version of the protocol required, etc.), as well as configuration of the TOE needed to communicate with other IT entities. The evaluator shall perform the following test for this requirement:~~

- ~~Test 1: The evaluator shall establish a session between an external IT entity and the TOE according to the configuration guidance provided. The evaluator shall then examine the traffic that passes between the IT entity and the TOE during several activities of the TOE. The evaluator shall observe that these data are not able to be viewed in the clear during this transfer, and that they are successfully received by the TOE. The evaluator shall perform this test for each protocol selected in the second selection.~~

TOE is not an audit server: The evaluator shall examine the TSS to ensure it describes the means by which the audit data are transferred to the external audit server, and how the trusted channel is provided. Testing of the trusted channel mechanism will be performed as specified in the associated assurance activities for the particular trusted channel mechanism. The evaluator shall also examine the operational guidance to ensure it describes how to establish the trusted channel to the audit server, as well as describe any requirements on the audit server (particular audit server protocol, version of the protocol required, etc.), as well as configuration of the TOE needed to communicate with the audit server. The evaluator shall perform the following test for this requirement:

- Test 1: The evaluator shall establish a session between the TOE and the audit server according to the configuration guidance provided. The evaluator shall then examine the traffic that passes between the audit server and the TOE during several activities of the evaluator's choice designed to generate audit data to be transferred to the audit server. The evaluator shall observe that these data are not able to be viewed in the clear during this transfer, and that they are successfully received by the audit server. The evaluator shall record the particular software (name, version) used on the audit server during testing.

6.1.2 Cryptographic Support (FCS)

6.1.2.1 FCS_CKM.1(1) Cryptographic Key Generation (for asymmetric keys)

FCS_CKM.1.1(1) The TSF 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 elliptic curve-based key establishment schemes and implementing "NIST curves" P-256, P-384 and no other curves (as defined in FIPS PUB 186-3, "Digital Signature Standard");
- NIST Special Publication 800-56A. "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" for finite field-based key establishment schemes;

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

EP Application Note: *The EP requires specific algorithms to be used in key establishment, and this instantiation of the requirement from the NDPP ensures the right selections are made.*

PP Application Note: *This component requires that the TOE be able to generate the public/private key pairs that are used for key establishment purposes for the various cryptographic protocols used by the TOE (e.g., IPsec). If multiple schemes are supported, then the ST author should iterate this requirement to capture this capability. The scheme used will be chosen by the ST author from the selection.*

Since the domain parameters to be used are specified by the requirements of the protocol in the NDPP, it is not expected that the TOE will generate domain parameters, and therefore there is no additional domain parameter validation needed when the TOE complies to the protocols specified in the NDPP.

SP 800-56B references (but does not mandate) key generation according to FIPS 186-3. For purposes of compliance in this version of the NDPP, RSA key pair generation according to FIPS 186-2 or FIPS 186-3 is allowed in order for the TOE to claim conformance to SP 800-56B.

The generated key strength of 2048-bit DSA and rDSA keys need to be 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.

EP Assurance Activity:

TSS: In order to show that the TSF complies with 800-56A and 800-56B (as selected) depending on the selections made, the evaluator shall ensure that the TSS contains the following information:

- The TSS shall list all sections of the appropriate 800-56 standard(s) to which the TOE complies.
- For each applicable section listed in the TSS, for all statements that are not "shall" (that is, "shall not", "should", and "should not"), if the TOE implements such options it shall be described in the TSS. If the included functionality is indicated as "shall not" or "should not" in the standard, the TSS shall provide a rationale for why this will not adversely affect the security policy implemented by the TOE;
- For each applicable section of 800-56A and 800-56B (as selected), any omission of functionality related to "shall" or "should" statements shall be described;

Any TOE-specific extensions, processing that is not included in the documents, or alternative implementations allowed by the documents that may impact the security requirements the TOE is to enforce shall be described.

Guidance: The evaluator shall check that the operational guidance describes how the key generation functionality is invoked, and describes the inputs and outputs associated with the process for each signature scheme supported. The evaluator shall also check that guidance is provided regarding the format and location of the output of the key generation process.

Test: The evaluator shall use the key pair generation portions of "The FIPS 186-3 Digital Signature Algorithm Validation System (DSA2VS)", "The FIPS 186-3 Elliptic Curve Digital Signature Algorithm Validation System (ECDSA2VS)", and "The RSA Validation System (RSA2VS)" as a guide in testing the requirement above, depending on the selection performed by the ST author. This will require that the evaluator have a trusted reference implementation of the algorithms that can produce test vectors that are verifiable during the test.

PP Assurance Activity: The evaluator shall use the key pair generation portions of "The FIPS 186-3 Digital Signature Algorithm Validation System (DSA2VS)", "The FIPS 186-3 Elliptic Curve Digital Signature Algorithm Validation System (ECDSA2VS)", and either "The RSA Validation System (RSAVS)" (for FIPS 186-2) or "The 186-3 RSA Validation System (RSA2VS)" (for FIPS 186-3) as a guide in testing the requirement above, depending on the selection performed by the ST author. This will require that the evaluator have a trusted reference

implementation of the algorithms that can produce test vectors that are verifiable during the test.

The evaluator shall ensure that the TSS contains a description of how the TSF complies with 800-56A and/or 800-56B, depending on the selections made. This description shall indicate the sections in 800-56A and/or 800-56B that are implemented by the TSF, and the evaluator shall ensure that key establishment is among those sections that the TSF claims to implement.

Any TOE-specific extensions, processing that is not included in the documents, or alternative implementations allowed by the documents that may impact the security requirements the TOE is to enforce shall be described.

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

FCS_CKM.1.2 The TSF shall generate asymmetric cryptographic keys used for IKE peer authentication in accordance with a:

- FIPS PUB 186-3, "Digital Signature Standard (DSS)", Appendix B.4 for ECDSA schemes and implementing "NIST curves" P-256, P-384 and no other curves;
- ANSI X9.31-1998, ~~***Appendix A.2.4 Using AES for RSA schemes***~~ ***Section 4.1***[2]

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

EP Application Note: *The ANSI X9.31-1998 option will be removed from the selection in a future publication of the EP. Presently, the selection is not exclusively limited to the FIPS PUB 186-3 options in order to allow industry some further time to complete the transition to the modern FIPS PUB 186-3 standard.*

The keys that are required to be generated by the TOE through this requirement are intended to be used for the authentication of the VPN peers during the IKE (either v1 or v2) key exchange. While it is required that the public key be associated with an identity in an X509v3 certificate, this association is not required to be performed by the TOE, and instead is expected to be performed by a Certificate Authority in the Operational Environment.

As indicated in FCS_IPSEC_EXT.1 - .1 Extended: Internet Protocol Security (IPsec) Communications, the TOE is required to implement support RSA or ECDSA (or both) for peer authentication.

The generated key strength of 2048-bit RSA keys need to be equivalent to, or greater than, a symmetric key strength of 112 bits. See NIST Special

[2] Refined according to CCEVS TD0031.

Publication 800-57, "Recommendation for Key Management" for information about equivalent key strengths.

EP Assurance Activity:

TSS: The evaluator shall check to ensure that the TSS describes how the key-pairs are generated. In order to show that the TSF implementation complies with FIPS PUB 186-3, the evaluator shall ensure that the TSS contains the following information:

- The TSS shall list all sections of Appendix B to which the TOE complies.
- For each applicable section listed in the TSS, for all statements that are not "shall" (that is, "shall not", "should", and "should not"), if the TOE implements such options it shall be described in the TSS. If the included functionality is indicated as "shall not" or "should not" in the standard, the TSS shall provide a rationale for why this will not adversely affect the security policy implemented by the TOE;
- For each applicable section of Appendix B, any omission of functionality related to "shall" or "should" statements shall be described;

Any TOE-specific extensions, processing that is not included in the Appendices, or alternative implementations allowed by the Appendices that may impact the security requirements the TOE is to enforce shall be described.

Guidance: The evaluator shall check that the operational guidance describes how the key generation functionality is invoked, and describes the inputs and outputs associated with the process for each signature scheme supported. The evaluator shall also check that guidance is provided regarding the format and location of the output of the key generation process.

Test: The evaluator shall use the key pair generation portions of "The FIPS 186-3 Elliptic Curve Digital Signature Algorithm Validation System (ECDSA2VS)" and "The RSA Validation System (RSA2VS)" as a guide in testing the requirement above, depending on the selection performed by the ST author. This will require that the evaluator have a trusted reference implementation of the algorithms that can produce test vectors that are verifiable during the test.

6.1.2.3 FCS_CKM_EXT.4 Cryptographic Key Zeroization

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

PP Application Note: *"Cryptographic Critical Security Parameters" are defined in FIPS 140-2 as "security-related information (e.g., secret and private cryptographic keys, and authentication data such as passwords and PINs) whose disclosure or modification can compromise the security of a cryptographic module."*

The zeroization indicated above applies to each intermediate storage area for plaintext key/cryptographic critical security parameter (i.e., any storage, such as memory buffers, that is included in the path of such data) upon the transfer of the key/cryptographic critical security parameter to another location.

Assurance Activity

The evaluator shall check to ensure the TSS describes each of the secret keys (keys used for symmetric encryption), private keys, and CSPs used to generate key; when they are zeroized (for example, immediately after use, on system shutdown, etc.); and the type of zeroization procedure that is performed (overwrite with zeros, overwrite three times with random pattern, etc.). If different types of memory are used to store the materials to be protected, the evaluator shall check to ensure that the TSS describes the zeroization procedure in terms of the memory in which the data are stored (for example, "secret keys stored on flash are zeroized by overwriting once with zeros, while secret keys stored on the internal hard drive are zeroized by overwriting three times with a random pattern that is changed before each write").

6.1.2.4 FCS_COP.1(1) Cryptographic Operation (for data encryption/decryption)

FCS_COP.1.1(1)

The TSF shall perform encryption and decryption in accordance with a specified cryptographic algorithm AES operating in GCM, CBC and 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

EP Application Note:

The EP requires the modes GCM and CBC to be used in the IPsec and IKE protocols (FCS_IPSEC_EXT.1.4, FCS_IPSEC_EXT.1.6). Therefore, the FCS_COP.1.1(1) element in the NDPP has been specified here to ensure the ST Author includes these two modes to be consistent with the IPsec requirements.

PP Application Note:

For the first selection, the ST author should choose the mode or modes in which AES operates to support the cryptographic protocols chosen for FTP_ITC and FTP_TRP. If any other modes are used to support requirements in the ST, those should be filled in through the assignment. For the second selection, the ST author should choose the standards that describe the modes specified in the first selection and the assignment.

Assurance Activity:

The evaluator shall use tests appropriate to the modes selected in the above requirement from "The Advanced Encryption Standard Algorithm Validation Suite (AESAVS)", "The XTS-AES Validation System (XTSVS)", The CMAC Validation System (CMACVS)", "The Counter with Cipher Block Chaining-Message Authentication Code (CCM) Validation System (CCMVS)", and "The Galois/Counter Mode (GCM) and GMAC Validation System (GCMVS)" (these documents are available from http://csrc.nist.gov/groups/STM/cavp/index.html) as a guide in testing the requirement above. This will require that the evaluator have a reference

implementation of the algorithms known to be good that can produce test vectors that are verifiable during the test.

6.1.2.5 FCS_COP.1(2) Cryptographic Operations (for cryptographic signature)

FCS_COP.1.1(2) The TSF shall perform cryptographic signature services in accordance with a

- RSA Digital Signature Algorithm (RSA) with a key size (modulus) of 2048 bits or greater that meets the following:
  - FIPS PUB 186-2 or FIPS PUB 186-3, "Digital Signature Standard"
- Elliptic Curve Digital Signature Algorithm (ECDSA) with a key size of 256 bits or greater that meets the following:
  - FIPS PUB 186-3, "Digital Signature Standard"
  - The TSF shall implement "NIST curves" P-256, P-384 and no other curves (as defined in FIPS PUB 186-3, "Digital Signature Standard").

PP Application Note: *As the preferred approach for cryptographic signature, elliptic curves will be required in future publications of the NDPP.*

PP Application Note: *The ST Author should choose the algorithm implemented to perform digital signatures; if more than one algorithm is available, this requirement (and the corresponding FCS_CKM.1 requirement) should be iterated to specify the functionality. For the algorithm chosen, the ST author should make the appropriate assignments/selections to specify the parameters that are implemented for that algorithm.*

For elliptic curve-based schemes, the key size refers to the log_2 of the order of the base point. As the preferred approach for digital signatures, ECDSA will be required in future publications of the NDPP.

Assurance Activity The evaluator shall use the signature generation and signature verification portions of "The Digital Signature Algorithm Validation System" (DSA2VS), "The Elliptic Curve Digital Signature Algorithm Validation System" (ECDSA2VS), and "The RSA Validation System" (RSAVS (for 186-2) or RSA2VS (for 186-3)) as a guide in testing the requirement above. The Validation System used shall comply with the conformance standard identified in the ST (i.e., FIPS PUB 186-2 or FIPS PUB 186-3). This will require that the evaluator have a reference implementation of the algorithms known to be good that can produce test vectors that are verifiable during the test.

6.1.2.6 FCS_COP.1(3) Cryptographic Operation (for cryptographic hashing)

FCS_COP.1.1(3) The TSF shall perform cryptographic hashing services in accordance with a specified cryptographic algorithm SHA-1, SHA-224, SHA-256, SHA-384, SHA-512 and message digest sizes 160, 224, 256, 384, 512 bits that meet the following: FIPS Pub 180-3, "Secure Hash Standard."

PP Application Note: *The selection of the hashing algorithm must correspond to the selection of the message digest size; for example, if SHA-1 is chosen, then the only valid message digest size selection would be 160 bits.*

In subsequent publications of the NDPP, it is likely that SHA-1 will no longer be an approved algorithm for cryptographic hashing.

Assurance Activity: The evaluator shall use "The Secure Hash Algorithm Validation System (SHAVS)" as a guide in testing the requirement above. This will require that the evaluator have a reference implementation of the algorithms known to be good that can produce test vectors that are verifiable during the test.

6.1.2.7 FCS_COP.1(4) Cryptographic Operation (for keyed hash message authentication)

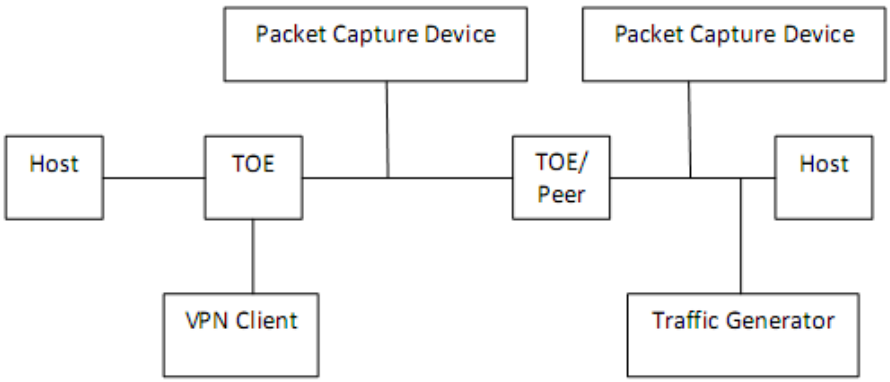
FCS_COP.1.1(4) The TSF shall perform keyed-hash message authentication in accordance with a specified cryptographic algorithm HMAC-SHA-1, SHA-224, SHA-256, SHA-384, SHA-512, key size **160, 224, 256, 384, 512 bits**, and message digest sizes 160, 224, 256, 384, 512 bits that meet the following: FIPS Pub 198-1, "The Keyed-Hash Message Authentication Code, and FIPS Pub 180-3, "Secure Hash Standard."

PP Application Note: *In future version of the NDPP, SHA-1 may be removed as a valid hash algorithm. Developers are encouraged to transition to the other listed hash algorithms.*

Assurance Activity: The evaluator shall use "The Keyed-Hash Message Authentication Code (HMAC) Validation System (HMACVS)" as a guide in testing the requirement above. This will require that the evaluator have a reference implementation of the algorithms known to be good that can produce test vectors that are verifiable during the test.

6.1.2.8 FCS_IPSEC_EXT.1 IPsec

Assurance Activity: In order to show that the TSF implements the RFCs correctly, the evaluator shall perform the assurance activities listed below. In future versions of the EP, assurance activities may be augmented, or new ones introduced that cover more aspects of RFC compliance than is currently described in the EP.



The evaluators shall minimally create a test environment equivalent to the test environment illustrated above. Two instantiations of the TOE will more than likely make it easier to conduct testing and if there is a failure of a test it should be more easily traced to the TOE, however, the evaluator is free to

construct a testbed where one instance of a TOE exists and there is a device that provides the necessary functions to interact with the TOE to satisfy the testing activities. If the ST author includes the requirements for a VPN Headend, it is expected that a VPN client be used to demonstrate the TOE can act as a remote access VPN headend as well as the requirements specified for VPN client management. It is expected that the traffic generator is used to construct network packets and will provide the evaluator with the ability manipulate fields in the ICMP, IPv4, ~~IPv6~~, UDP, and TCP packet headers. The evaluators must provide justification for any differences in the test environment. One such justification may be that the host can implement a traffic generator. It would be more difficult to make the same argument for the packet capture device, since it is expected the evaluator will have access to packets that are actually on the wire.

FCS_IPSEC_EXT.1.1 The TSF shall implement the IPsec architecture as specified in RFC 4301.

Assurance Activity[3]:

TSS: The evaluator shall examine the TSS and determine that it describes the rules for processing both inbound and outbound packets in terms of the IPsec policy. As noted in section 4.4.1 of RFC 4301, the processing of entries in the SPD is non-trivial and the evaluator shall determine that the description in the TSS is sufficient to determine which rules will be applied given the rule structure implemented by the TOE. For example, if the TOE allows specification of ranges, conditional rules, etc., the evaluator shall determine that the description of rule processing (for both inbound and outbound packets) is sufficient to determine the action that will be applied, especially in the case where two different rules may apply (for example, there may be a specific rule that specifies PROTECT, and a general rule that would apply to the same packet that specifies BYPASS). This description shall cover both the initial packets (that is, no SA is established on the interface or for that particular packet) as well as packets that are part of an established SA.

Guidance: The evaluator shall examine the operational guidance to verify it instructs the Administrator how to construct entries into the SPD that specify a rule for DISCARD, BYPASS and PROTECT. The evaluator shall determine that the description in the operational guidance is consistent with the description in the ST, and that the level of detail in the operational guidance is sufficient to allow the administrator to set up the SPD in an unambiguous fashion.

Test: The evaluator uses the operational guidance to configure the TOE and platform to carry out the following tests:

- Test 1: The evaluator shall configure the SPD such that there is a rule for DISCARD, BYPASS, PROTECT. The selectors used in the construction of the rule shall be different such that the evaluator

[3] The assurance activity was updated according to input from the VPNGW_TRRT on January 1, 2015.

can send in three network packets with the appropriate fields in the packet header that each packet will match one of the three rules. The evaluator observes via the audit trail, and packet captures that the TOE exhibited the expected behavior: appropriate packet was dropped, allowed through without modification, was encrypted by the IPsec implementation.

- Test 2: The evaluator shall devise several tests that cover a variety of scenarios for packet processing. These scenarios must exercise the range of possibilities for SPD entries and processing modes as outlined in the ST. Potential areas to cover include rules with overlapping ranges and conflicting entries, inbound and outbound packets, and packets that establish SAs as well as packets that belong to established SAs. The evaluator shall verify, for each scenario, that the expected behavior is exhibited, and is consistent with both the ST and the operational guidance.

FCS_IPSEC_EXT.1.2 The TSF shall implement tunnel mode.

EP Application Note: *Future versions of the EP will require that the TSF implement both tunnel mode and transport mode.*

Assurance Activity:

TSS: The evaluator checks the TSS to ensure it states that the VPN can be established to operate in tunnel mode and/or transport mode (as selected).

Guidance: The evaluator shall confirm that the operational guidance instructs the Administrator how the TOE is configured in each mode selected.

Test:

- Test 1 (conditional): If tunnel mode is selected, the evaluator uses the operational guidance to configure the TOE in tunnel mode, and a peer TOE in tunnel mode. The evaluator configures the two peer TOEs to use any of the allowable cryptographic algorithms, authentication methods, etc. to ensure an allowable SA can be negotiated. The evaluator shall then initiate a session between the peers. The evaluator observes in the audit trail and the captured packets that a successful connection was established using the tunnel mode.
- Test 2 (conditional): If transport mode is selected, the evaluator uses the operational guidance to configure the TOE to operate in transport mode when it receives packets from the VPN client. The evaluator configures the TOE and VPN client to use any of the allowed cryptographic algorithms, authentication methods, etc. to ensure an allowable SA can be negotiated. The evaluator then initiates a connection with the TOE using the VPN client. The evaluator observes in the audit trail and the captured packets that a successful connection was established using the transport mode.

FCS_IPSEC_EXT.1.3 The TSF shall have a nominal, final entry in the ~~***SPD***~~ ***iptables chain*** that matches anything that is otherwise unmatched, and discards it.

TOE Application Note: *iptables provides equivalent functionality to IPsec SPDs; however, iptables is an independent implementation. The VPNGW TRRT approved substitution of iptables for SPDs.*

Assurance Activity:

TSS: The evaluator shall examine the TSS to verify that the TSS provides a description of how a packet is processed against the SPD and that if no "rules" are found to match, that a final rule exists, either implicitly or explicitly, that causes the network packet to be discarded.

Guidance: The evaluator checks that the operational guidance provides instructions on how to construct the SPD and uses the guidance to configure the TOE for the following tests.

Test:

- Test 1: The evaluator shall configure the TOE's SPD, such that it has entries that contain operations that DISCARD, BYPASS, and PROTECT network packets. The evaluator also configures the TOE so that all auditable events with respect to FCS_IPSEC_EXT.1 are enabled. The evaluator may use the SPD that was created for verification of FCS_IPSEC_EXT.1.1. The evaluator shall construct a network packet that matches a BYPASS entry and send that packet to the TOE. The evaluator should observe that the network packet is passed to the proper destination interface with no modification. The evaluator shall then modify a field in the packet header; such that it no longer matches the evaluator-created entries (there may be a "TOE created" final entry that discards packets that do not match any previous entries). The evaluator sends the packet, and observes that the packet was not permitted to flow to any of the TOE's interfaces. The evaluator shall verify that an audit record is generated that specifies that the packet was discarded as expected.

FCS_IPSEC_EXT.1.4 The TSF 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.

EP Application Note: *If an AES-CBC selection is made, the SHA-based HMAC must be consistent with what is specified in the NDPP FCS_COP.1(4) Cryptographic Operation (for keyed-hash message authentication) requirement.*

Assurance Activity:

TSS: The evaluator shall examine the TSS to verify that the algorithms AES-GCM-128 and AES-GCM-256 are implemented. If the ST author has selected either AES-CBC-128 or AES-CBC-256 in this requirement, then the evaluator verifies the TSS describes these as well. In addition, the evaluator ensures that the SHA-based HMAC algorithm conforms to the algorithms specified in

FCS_COP.1(4) Cryptographic Operations (for keyed-hash message authentication.

Guidance: The evaluator checks the operational guidance to ensure it provides instructions on how to configure the TOE to use the AES-GCM-128, and AES-GCM-256 algorithms, and if either AES-CBC-128 or AES-CBC-256 have been selected the guidance instructs how to use these as well.

Test:

- Test 1: The evaluator shall configure the TOE as indicated in the operational guidance configuring the TOE to using each of the AES-GCM-128, and AES-GCM-256 algorithms, and attempt to establish a connection using ESP in confidentiality and integrity mode. If the ST Author has selected either AES-CBC-128 or AES-CBC-256, the TOE is configured to use those algorithms and the evaluator attempts to establish a connection using ESP in confidentiality and integrity mode for those algorithms selected.

FCS_IPSEC_EXT.1.5 The TSF shall implement the protocol: IKEv2 as defined in RFCs 5996 (with mandatory support for NAT traversal as specified in section 2.23) and RFC 4868 for hash functions.

PP Application Note: *Either IKEv1 or IKEv2 support must be provided, although conformant TOEs can provide both; the first selection is used to make this choice. For IKEv1, the requirement is to be interpreted as requiring the IKE implementation conforming to RFC 2409 with the additions/modifications as described in RFC 4109. RFC 4304 identifies support for extended sequence numbers, which compliant TOEs can specify using the second selection. RFC 4868 identifies additional hash functions for use with both IKEv1 and IKEv2; if these functions are implemented, the third (for IKEv1) and fourth (for IKEv2) selection can be used.*

Assurance Activity:

TSS: The evaluator shall examine the TSS to verify that IKEv1 and/or IKEv2 are implemented.

Guidance: The evaluator checks the operational guidance to ensure it instructs the administrator how to configure the TOE to use IKEv1 and/or IKEv2 (as selected), and uses the guidance to configure the TOE to perform NAT traversal for the following test.

Test:

- Test 1: The evaluator shall configure the TOE so that it will perform NAT traversal processing as described in the TSS and RFC 5996, section 2.23. The evaluator shall initiate an IPsec connection and determine that the NAT is successfully traversed.

FCS_IPSEC_EXT.1.6 The TSF 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 no other algorithm.

Assurance Activity:

TSS: The evaluator shall ensure the TSS identifies the algorithms used for encrypting the IKEv1 and/or IKEv2 payload, and that the algorithms AES-CBC-128, AES-CBC-256 are specified, and if others are chosen in the selection of the requirement, those are included in the TSS discussion.

Guidance: The evaluator ensures that the operational guidance describes how the TOE can be configured to use the mandated algorithms, as well as any additional algorithms selected in the requirement. The guidance is then used to configure the TOE to perform the following test.

Test:

- Test 1: The evaluator shall configure the TOE to use AES-CBC-128 to encrypt the IKEv1 and/or IKEv2 payload and establish a connection with a peer device, which is configured to only accept the payload encrypted using AES-CBC-128. The evaluator will consult the audit trail to confirm the algorithm was that used in the negotiation.

~~FCS_IPSEC_EXT.1.7 The TSF shall ensure that IKEv1 Phase 1 exchanges use only main mode.~~

~~*EP Application Note: Element 1.7 is only applicable if IKEv1 is selected.*~~

~~Assurance Activity:~~

~~TSS: The evaluator shall examine the TSS to ensure that, in the description of the IPsec protocol supported by the TOE, it states that aggressive mode is not used for IKEv1 Phase 1 exchanges, and that only main mode is used. It may be that this is a configurable option.~~

~~Guidance: If the mode requires configuration of the TOE prior to its operation, the evaluator shall check the operational guidance to ensure that instructions for this configuration are contained within that guidance.~~

~~Test:~~

- ~~Test 1 (conditional): The evaluator shall configure the TOE as indicated in the operational guidance, and attempt to establish a connection using an IKEv1 Phase 1 connection in aggressive mode. This attempt should fail. The evaluator should then show that main mode exchanges are supported. This test is not applicable if IKEv1 is not selected above in the FCS_IPSEC_EXT.1.5 protocol selection.~~

FCS_IPSEC_EXT.1.8 The TSF shall ensure that IKEv2 SA lifetimes can be configured by an Administrator based on number of ~~***packets***~~ ***bytes*** or length of time, where the time values can be limited to: 24 hours for Phase 1 SAs and 8 hours for Phase 2 SAs.

EP Application Note: It is appropriate to refine the requirement in terms of number of MB/KB instead of number of packets, as long as the TOE is capable of setting a limit on the amount of traffic that is protected by the same key (the total volume of all IPsec traffic protected by that key).

PP Application Note: *The ST Author is afforded a selection based on the version of IKE in their implementation. If the lifetime limitations are configurable, then the evaluator verifies that the appropriate instructions for configuring these values are included in the operational guidance.*

As far as SA lifetimes are concerned, the TOE can limit the lifetime based on the number of bytes transmitted, or the number of packets transmitted. Either packet-based or volume-based SA lifetimes are acceptable; the ST author makes the appropriate selection to indicate which type of lifetime limits are supported.

Assurance Activity:

TSS: How the lifetimes are established and enforced is described in the RFCs and the evaluator examines the TSS as stated at the beginning of this section.

Guidance: The evaluator verifies that the values for SA lifetimes can be configured and that the instructions for doing so are located in the operational guidance. The evaluator ensures that the Administrator is able to configurable Phase 1 SAs values for 24 hours and 8 hours for Phase 2 SAs. Currently there are no values mandated for the number of packets, the evaluator just ensures that this can be configured. The TOE may limit the lifetime on the number of bytes that have been transmitted and this would be acceptable.

Test: When testing this functionality, the evaluator needs to ensure that both sides are configured appropriately. From the RFC "A difference between IKEv1 and IKEv2 is that in IKEv1 SA lifetimes were negotiated. In IKEv2, each end of the SA is responsible for enforcing its own lifetime policy on the SA and rekeying the SA when necessary. If the two ends have different lifetime policies, the end with the shorter lifetime will end up always being the one to request the rekeying. If the two ends have the same lifetime policies, it is possible that both will initiate a rekeying at the same time (which will result in redundant SAs). To reduce the probability of this happening, the timing of rekeying requests SHOULD be jittered."

Each of the following tests shall be performed for each version of IKE selected in the FCS_IPSEC_EXT.1.5 protocol selection

- Test 1: The evaluator shall configure a maximum lifetime in terms of the # of packets (or bytes) allowed following the operational guidance. The evaluator shall establish an SA and determine that once the allowed # of packets (or bytes) through this SA is exceeded, the connection is closed.
- Test 2: The evaluator shall construct a test where a Phase 1 SA is established and attempted to be maintained for more than 24 hours before it is renegotiated. The evaluator shall observe that this SA is closed or renegotiated in 24 hours or less. If such an action requires that the TOE be configured in a specific way, the evaluator shall implement tests demonstrating that the configuration capability of the TOE works as documented in the operational guidance.

- Test 3: The evaluator shall perform a test similar to Test 1 for Phase 2 SAs, except that the lifetime will be 8 hours instead of 24.

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

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

Assurance Activity: The evaluator shall check to ensure that, for each DH group supported by the TSF, the TSS describes the process for generating "x" (as defined in FCS_IPSEC_EXT.1.9) and each nonce. The evaluator shall verify that the TSS indicates that the random number generated that meets the requirements in this PP is used, and that the length of "x" and the nonces meet the stipulations in the requirement.

FCS_IPSEC_EXT.1.11 The TSF 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).

PP Application Note: *The above requires that the TOE support DH Group 14. If other groups are supported, then those should be selected (for groups 24, 19, 20, and 5) or specified in the assignment above; otherwise, "no other DH groups" should be selected. This applies to IKEv1/IKEv2 exchanges.*

In future publications of the NDPP DH Groups 19 (256-bit Random ECP) and 20 (384-bit Random ECP) will be required.

Assurance Activity: The evaluator shall check to ensure that the DH groups specified in the requirement are listed as being supported in the TSS. If there is more than one DH group supported, the evaluator checks to ensure the TSS describes how a particular DH group is specified/negotiated with a peer. The evaluator shall also perform the following test:

- Test 1: For each supported DH group, the evaluator shall test to ensure that all IKE protocols can be successfully completed using that particular DH group.

FCS_IPSEC_EXT.1.12 The TSF 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.

PP Application Note: *The selected algorithm should correspond to an appropriate selection for* FCS_COP.1(2)*. If IPsec is included in the TOE, the ST author also includes FIA_PSK_EXT from Appendix C.*

Assurance Activity:

TSS: The evaluator ensures that the TSS identifies RSA and/or ECDSA as being used to perform peer authentication. The description must be consistent with the algorithms specified in FCS_COP.1(2) Cryptographic Operations (for cryptographic signature).

Guidance: The evaluator ensures the operational guidance describes how to set up the TOE to use the cryptographic algorithms RSA and/or ECDSA.

In order to construct the environment and configure the TOE for the following tests, the evaluator will ensure that the operation guidance also describes how to configure the TOE to connect to a trusted CA, and ensure a valid certificate for that CA is loaded into the TOE and marked "trusted".

Test: For efficiency sake, the testing that is performed here has been combined with aspects of the testing for FIA_X509_EXT.1 Extended: X.509 Certificates, specifically FIA_X509_EXT.1.4, and FIA_X509_EXT.1.5.

The following five tests shall be repeated for each peer authentication protocol selected in the FCS_IPSEC_EXT.1.12 selection above:

- Test 1: The evaluator shall have the TOE generate a public-private key pair, and submit a CSR (Certificate Signing Request) to a CA (trusted by both the TOE and the peer VPN used to establish a connection) for its signature. The values for the DN (Common Name, Organization, Organizational Unit, and Country) will also be passed in the request.
- Test 2: The evaluator shall use a certificate signed using the RSA or ECDSA algorithm to authenticate the remote peer during the IKE exchange. This test ensures the remote peer has the certificate for the trusted CA that signed the TOE's certificate and it will do a bit-wise comparison on the DN. This bit-wise comparison of the DN ensures that not only does the peer have a certificate signed by the trusted CA, but the certificate is from the DN that is expected. The evaluator will configure the TOE to associate a certificate (e.g., a certificate map in some implementations) with a VPN connection. This is what the DN is checked against.
- Test 3: The evaluator shall test that the TOE can properly handle revoked certificates – conditional on whether CRL or OCSP is selected; if both are selected, and then a test is performed for each method. For this draft of the EP, the evaluator has to only test one up in the trust chain (future drafts may require to ensure the validation is done up the entire chain). The evaluator shall ensure that a valid certificate is used, and that the SA is established. The evaluator then attempts the test with a certificate that will be revoked (for each method chosen in the selection) to ensure when the certificate is no longer valid that the TOE will not establish an SA.
- Test 4: The evaluator shall construct a certificate path, such that the certificate of the CA issuing the TOE's certificate does not contain the basicConstraints extension. The validation of the certificate path fails.
- Test 5: The evaluator shall construct a certificate path, such that the certificate of the CA issuing the TOE's certificate has the cA flag in

the basicConstraints extension not set. The validation of the certificate path fails.

- Test 6: The evaluator shall construct a certificate path, such that the certificate of the CA issuing the TOE's certificate has the cA flag in the basicConstraints extension set to TRUE. The validation of the certificate path succeeds.
- Test 7: The evaluator shall test that given a signed certificate from a trusted CA, that when the DN does not match – any of the four fields can be modified such that they do not match the expected value, that an SA does not get established.
- Test 8: The evaluator shall ensure that the TOE is configurable to either establish an SA, or not establish an SA if a connection to the certificate validation entity cannot be reached. For each method selected for certificate validation, the evaluator attempts to validate the certificate – for the purposes of this test, it does not matter if the certificate is revoked or not. For the "mode" where an SA is allowed to be established, the connection is made. Where the SA is not to be established, the connection is refused.

FCS_IPSEC_EXT.1.13 The TSF 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.

Assurance Activity:

TSS: The evaluator shall check that the TSS describes the potential strengths (in terms of the number of bits in the symmetric key) of the algorithms that are allowed for the IKE and ESP exchanges. The TSS shall also describe the checks that are done when negotiating IKEv1 Phase 2 and/or IKEv2 CHILD_SA suites to ensure that the strength (in terms of the number of bits of key in the symmetric algorithm) of the negotiated algorithm is less than or equal to that of the IKE SA this is protecting the negotiation.

Guidance: The evaluator simply follows the guidance to configure the TOE to perform the following tests.

Test:

- Test 1: This test shall be performed for each version of IKE supported by the TOE. The evaluator shall successfully negotiate an IPsec connection using each of the supported algorithms and hash functions identified in the requirements.
- Test 2: This test shall be performed for each version of IKE supported by the TOE. The evaluator shall attempt to establish an SA for ESP that selects an encryption algorithm with more strength than that being used for the IKE SA (i.e., symmetric algorithm with a

key size larger than that being used for the IKE SA). Such attempts should fail.

- Test 3: This test shall be performed for each version of IKE supported by the TOE. The evaluator shall attempt to establish an IKE SA using an algorithm that is not one of the supported algorithms and hash functions identified in the requirements. Such an attempt should fail.
- Test 4: This test shall be performed for each version of IKE supported by the TOE. The evaluator shall attempt to establish an SA for ESP (assumes the proper parameters where used to establish the IKE SA) that selects an encryption algorithm that is not identified in FCS_IPSEC_EXT.1.4. Such an attempt should fail.

6.1.2.9 FCS_TLS_EXT.1 TLS

FCS_TLS_EXT.1.1 The TSF shall implement one or more of the following protocols TLS 1.2 (RFC 5246) supporting the following ciphersuites:

Mandatory Ciphersuites:

- TLS_RSA_WITH_AES_128_CBC_SHA

Optional Ciphersuites:

- TLS_RSA_WITH_AES_256_CBC_SHA

PP Application Note: *The ST author must make the appropriate selections and assignments to reflect the TLS implementation.*

The ciphersuites to be tested in the evaluated configuration are limited by this requirement. The ST author should select the optional ciphersuites that are supported; if there are no ciphersuites supported other than the mandatory suites, then "None" should be selected. If administrative steps need to be taken so that the suites negotiated by the implementation are limited to those in this requirement, the appropriate instructions need to be contained in the guidance called for by AGD_OPE.

The Suite B algorithms (RFC 5430) listed above are the preferred algorithms for implementation. The TLS requirement will be changed in the next version of the NDPP to comply with NIST SP 800-131A.

Assurance Activity: The evaluator shall check the description of the implementation of this protocol in the TSS to ensure that the ciphersuites supported are specified. The evaluator shall check the TSS to ensure that the ciphersuites specified are identical to those listed for this component. The evaluator shall also check the operational guidance to ensure that it contains instructions on configuring the TOE so that TLS conforms to the description in the TSS (for instance, the set of ciphersuites advertised by the TOE may have to be restricted to meet the requirements). The evaluator shall also perform the following test:

- Test 1: The evaluator shall establish a TLS connection using each of the ciphersuites specified by the requirement. This connection may be established as part of the establishment of a higher-level protocol, e.g., as part of a HTTPS session. It is sufficient to observe the successful negotiation of a ciphersuite to satisfy the intent of the test; it is not necessary to examine the characteristics of the encrypted traffic in an attempt to discern the ciphersuite being used (for example, that the cryptographic algorithm is 128-bit AES and not 256-bit AES).
- ~~***Test 2: The evaluator shall setup a man-in-the-middle tool between the TOE and the TLS Peer and shall perform the following modifications to the traffic:***~~
    - ~~***[Conditional: TOE is a server] Modify at least one byte in the server's nonce in the Server Hello handshake message, and verify that the server denies the client's Finished handshake message.***~~
    - ~~***[Conditional: TOE is a client] Modify the server's selected ciphersuite in the Server Hello handshake message to be a ciphersuite not presented in the Client Hello handshake message. The evaluator shall verify that the client rejects the connection after receiving the Server Hello.***~~
    - ~~***[Conditional: TOE is a client] If a DHE or ECDHE ciphersuite is supported, modify the signature block in the Server's KeyExchange handshake message, and verify that the client rejects the connection after receiving the Server KeyExchange.***~~
    - ~~***[Conditional: TOE is a client] Modify a byte in the Server Finished handshake message, and verify that the client sends a fatal alert upon receipt and does not send any application data.***~~[4]

6.1.2.10 FCS_SSH_EXT.1 SSH

FCS_SSH_EXT.1.1 The TSF shall implement the SSH protocol that complies with RFCs 4251, 4252, 4253, 4254, and no other RFCs.

PP Application Note: *The ST author selects which of the additional RFCs to which conformance is being claimed. Note that these need to be consistent with selections in later elements of this component (e.g., cryptographic algorithms permitted).*

In the next version of the NDPP, a requirement will be added regarding rekeying. The requirement will read "The TSF shall ensure that the SSH connection be rekeyed after no more than 2^{28} packets have been transmitted using that key."

[4] Removed according to CCEVS TD0004.

FCS_SSH_EXT.1.2 The TSF shall ensure that the SSH protocol implementation supports the following authentication methods as described in RFC 4252: public key-based, password-based.

Assurance Activity: The evaluator shall check to ensure that the TSS contains a description of the public key algorithms that are acceptable for use for authentication, that this list conforms to FCS_SSH_EXT.1.5, and ensure that password-based authentication methods are also allowed. The evaluator shall also perform the following tests:

- Test 1: The evaluator shall, for each public key algorithm supported, show that the TOE supports the use of that public key algorithm to authenticate a user connection. Any configuration activities required to support this test shall be performed according to instructions in the operational guidance.
- Test 2: Using the operational guidance, the evaluator shall configure the TOE to accept password-based authentication, and demonstrate that a user can be successfully authenticated to the TOE over SSH using a password as an authenticator.

FCS_SSH_EXT.1.3 The TSF shall ensure that, as described in RFC 4253, packets greater than **32768** bytes in an SSH transport connection are dropped.

PP Application Note: *RFC 4253 provides for the acceptance of "large packets" with the caveat that packets should be of "reasonable length" or dropped. The assignment should be filled in by the ST author with the maximum packet size accepted, thus defining "reasonable length" for the TOE.*

Assurance Activity: The evaluator shall check that the TSS describes how "large packets" in terms of RFC 4253 are detected and handled. The evaluator shall also perform the following test:

- Test 1: The evaluator shall demonstrate that if the TOE receives a packet larger than that specified in this component, that packet is dropped.

FCS_SSH_EXT.1.4 The TSF shall ensure that the SSH transport implementation uses the following encryption algorithms: AES-CBC-128, AES-CBC-256, no other algorithms.

PP Application Note: *In the assignment, the ST author can select the AES-GCM algorithms, or "no other algorithms" if AES-GCM is not supported. If AES-GCM is selected, there should be corresponding FCS_COP entries in the ST.*

Assurance Activity: The evaluator shall check the description of the implementation of this protocol in the TSS to ensure that optional characteristics are specified, and the encryption algorithms supported are specified as well. The evaluator shall check the TSS to ensure that the encryption algorithms specified are identical to those listed for this component. The evaluator shall also check the operational guidance to ensure that it contains instructions on configuring the TOE so that SSH conforms to the description in the TSS (for instance, the set of algorithms advertised by the TOE may have to be

restricted to meet the requirements). The evaluator shall also perform the following test:

- Test 1: The evaluator shall establish a SSH connection using each of the encryption algorithms specified by the requirement. It is sufficient to observe (on the wire) the successful negotiation of the algorithm to satisfy the intent of the test.

FCS_SSH_EXT.1.5 The TSF shall ensure that the SSH transport implementation uses SSH_RSA and no other public key algorithms as its public key algorithm(s).

PP Application Note: *Implementations that select only SSH_RSA will not achieve the 112-bit security strength in the digital signature generation for SSH authentication as is recommended in NIST SP 800-131A. Future versions of this profile will likely disallow the option of selecting only SSH_RSA.*

Assurance Activity: The assurance activity associated with FCS_SSH_EXT.1.4 verifies this requirement.

FCS_SSH_EXT.1.6 The TSF shall ensure that data integrity algorithms used in SSH transport connection is hmac-sha1.

PP Application Note: *RFC 6668 specifies the use of the sha2 algorithms in SSH.*

Assurance Activity: The evaluator shall check the TSS to ensure that it lists the supported data integrity algorithms, and that that list corresponds to the list in this component. The evaluator shall also check the operational guidance to ensure that it contains instructions to the administrator on how to ensure that only the allowed data integrity algorithms are used in SSH connections with the TOE (specifically, that the "none" MAC algorithm is not allowed). The evaluator shall also perform the following test:

- Test 1: The evaluator shall establish a SSH connection using each of the integrity algorithms specified by the requirement. It is sufficient to observe (on the wire) the successful negotiation of the algorithm to satisfy the intent of the test.

FCS_SSH_EXT.1.7 The TSF shall ensure that diffie-hellman-group14-sha1 and no other methods are the only allowed key exchange method used for the SSH protocol.

Assurance Activity: The evaluator shall ensure that operational guidance contains configuration information that will allow the security administrator to configure the TOE so that all key exchanges for SSH are performed using DH group 14 and any groups specified from the selection in the ST. If this capability is "hard-coded" into the TOE, the evaluator shall check the TSS to ensure that this is stated in the discussion of the SSH protocol. The evaluator shall also perform the following test:

- Test 1: The evaluator shall attempt to perform a diffie-hellman-group1-sha1 key exchange, and observe that the attempt fails. For each allowed key exchange method, the evaluator shall then attempt to perform a key exchange using that method, and observe that the attempt succeeds.

6.1.2.11 FCS_RBG_EXT.1 Extended: Cryptographic Operation (Random Bit Generation)

FCS_RBG_EXT.1.1 The TSF shall perform all random bit generation (RBG) services in accordance with CTR_DRBG (AES) seeded by an entropy source that accumulated entropy from a TSF-hardware based noise source, and a software-based noise source.

FCS_RBG_EXT.1.2 The deterministic RBG shall be seeded with a minimum of 256 bits of entropy at least equal to the greatest security strength of the keys and hashes that it will generate.

EP Application Note: *The NDPP allows the ST Author to choose whether the noise source is software based or hardware based. For compliance with this EP, there must be at least one hardware based noise source.*

A hardware noise source is a component that produces data that cannot be explained by a deterministic rule, due to its physical nature. In other words, a hardware based noise source generates sequences of random numbers from a physical process that cannot be predicted. For example, a sampled ring oscillator consists of an odd number of inverter gates chained into a loop, with an electrical pulse traveling from inverter to inverter around the loop. The inverters are not clocked, so the precise time required for a complete circuit around the loop varies slightly as various physical effects modify the small delay time at each inverter on the line to the next inverter. This variance results in an approximate natural frequency that contains drift and jitter over time. The output of the ring oscillator consists of the oscillating binary value sampled at a constant rate from one of the inverters – a rate that is significantly slower than the oscillator's natural frequency.

Any hardware component behaving in similarly variable ways that cannot be explained by a precise and predictable rule can serve as a hardware-based noise source. It is also possible to use multiple independent noise sources to increase entropy production and reduce attack potential (by requiring attackers to exploit multiple random bit streams) as long as at least one of the sources is hardware based. It should be noted that timing of interrupts caused by mechanical I/O devices and system counters are not considered hardware-based noise sources for the purposes of this requirement.

See Appendix D of the NDPP for further explanation regarding entropy.

PP Application Note: *NIST Special Pub 800-90B describes the minimum entropy measurement that will probably be required future versions of FIPS-140. If possible this should be used immediately and will be required in future versions of the NDPP.*

For the first selection in FCS_RBG_EXT.1.1, the ST author should select the standard to which the RBG services comply (either 800-90B or 140-2 Annex C).

SP 800-90B contains four different methods of generating random numbers; each of these, in turn, depends on underlying cryptographic primitives (hash functions/ciphers). The ST author will select the function used (if 800-90B is selected), and include the specific underlying cryptographic primitives used

in the requirement or in the TSS. While any of the identified hash functions (SHA-1, SHA-224,SHA-256, SHA-384, SHA-512) are allowed for Hash_DRBG or HMAC_DRBG, only AES-based implementations for CTR_DRBG are allowed. While any of the curves defined in 800-90B are allowed for Dual_EC_DRBG, the ST author not only must include the curve chosen, but also the hash algorithm used.

For the second selection in FCS_RBG_EXT.1.1, the ST author indicates whether the sources of entropy are software-based, hardware-based, or both. If there are multiple sources of entropy, the ST will elaborate each entropy sources and whether it is hardware- or software-based. Hardware-based noise sources are preferred.

Note that for FIPS Pub 140-2 Annex C, currently only the method described in NIST-Recommended Random Number Generator Based on ANSI X9.31 Appendix A.2.4 Using the 3-Key Triple DES and AES Algorithms, Section 3 is valid. If the key length for the AES implementation used here is different than that used to encrypt the user data, then FCS_COP.1 may have to be adjusted or iterated to reflect the different key length. For the selection in FCS_RBG_EXT.1.2, the ST author selects the minimum number of bits of entropy that is used to seed the RBG.

The ST author also ensures that any underlying functions are included in the baseline requirements for the TOE.

For the selection in FCS_RBG_EXT.1.2, the ST author selects the appropriate number of bits of entropy that corresponds to the greatest security strength of the algorithms included in the ST. Security strength is defined in Tables 2 and 3 of NIST SP 800-57A. For example, if the implementation includes 2048-bit RSA (security strength of 112 bits), AES 128 (security strength 128 bits), and HMAC-512 (security strength 256 bits), then the ST author would select 256 bits.

Assurance Activity: Documentation shall be produced – and the evaluator shall perform the activities – in accordance with Annex D, Entropy Documentation and Assessment.

The evaluator shall also perform the following tests, depending on the standard to which the RBG conforms.

~~Implementations Conforming to FIPS 140-2, Annex C~~

~~The reference for the tests contained in this section is The Random Number Generator Validation System (RNGVS) [RNGVS]. The evaluator shall conduct the following two tests. Note that the "expected values" are produced by a reference implementation of the algorithm that is known to be correct. Proof of correctness is left to each Scheme.~~

~~The evaluator shall perform a Variable Seed Test. The evaluator shall provide a set of 128 (Seed, DT) pairs to the TSF RBG function, each 128 bits. The evaluator shall also provide a key (of the length appropriate to the AES algorithm) that is constant for all 128 (Seed, DT) pairs. The DT value is incremented by 1 for each set. The seed values shall have no repeats within~~

~~the set. The evaluator ensures that the values returned by the TSF match the expected values.~~

~~The evaluator shall perform a Monte Carlo Test. For this test, they supply an initial Seed and DT value to the TSF RBG function; each of these is 128 bits. The evaluator shall also provide a key (of the length appropriate to the AES algorithm) that is constant throughout the test. The evaluator then invokes the TSF RBG 10,000 times, with the DT value being incremented by 1 on each iteration, and the new seed for the subsequent iteration produced as specified in NIST-Recommended Random Number Generator Based on ANSI X9.31 Appendix A.2.4 Using the 3-Key Triple DES and AES Algorithms, Section 3. The evaluator ensures that the 10,000th value produced matches the expected value.~~

Implementations Conforming to NIST Special Publication 800-90A

The evaluator shall perform 15 trials for the RBG implementation. If the RBG is configurable, the evaluator shall perform 15 trials for each configuration. The evaluator shall also confirm that the operational guidance contains appropriate instructions for configuring the RBG functionality.

If the RBG has prediction resistance enabled, each trial consists of (1) instantiate DRBG, (2) generate the first block of random bits (3) generate a second block of random bits (4) uninstantiate. The evaluator verifies that the second block of random bits is the expected value. The evaluator shall generate eight input values for each trial. The first is a count (0 - 14). The next three are entropy input, nonce, and personalization string for the instantiate operation. The next two are additional input and entropy input for the first call to generate. The final two are additional input and entropy input for the second call to generate. These values are randomly generated. “generate one block of random bits” means to generate random bits with number of returned bits equal to the Output Block Length (as defined in NIST SP 800-90A).

If the RBG does not have prediction resistance, each trial consists of (1) instantiate DRBG, (2) generate the first block of random bits (3) reseed, (4) generate a second block of random bits (5) uninstantiate. The evaluator verifies that the second block of random bits is the expected value. The evaluator shall generate eight input values for each trial. The first is a count (0 - 14). The next three are entropy input, nonce, and personalization string for the instantiate operation. The fifth value is additional input to the first call to generate. The sixth and seventh are additional input and entropy input to the call to reseed. The final value is additional input to the second generate call.

The following paragraphs contain more information on some of the input values to be generated/selected by the evaluator.

- Entropy input: the length of the entropy input value must equal the seed length.
- Nonce: If a nonce is supported (CTR_DRBG with no df does not use a nonce), the nonce bit length is one-half the seed length.

- Personalization string: The length of the personalization string must be <= seed length. If the implementation only supports one personalization string length, then the same length can be used for both values. If more than one string length is supported, the evaluator shall use personalization strings of two different lengths. If the implementation does not use a personalization string, no value needs to be supplied.
- Additional input: the additional input bit lengths have the same defaults and restrictions as the personalization string lengths.

6.1.3 User Data Protection (FDP)

6.1.3.1 FDP_RIP.2 Full Residual Information Protection

FDP_RIP.2.1 The TSF shall ensure that any previous information content of a resource is made unavailable upon the allocation of the resource to, deallocation of the resource from all objects.

Assurance Activity: "Resources" in the context of this requirement are network packets being sent through (as opposed to "to", as is the case when a security administrator connects to the TOE) the TOE. The concern is that once a network packet is sent, the buffer or memory area used by the packet still contains data from that packet, and that if that buffer is re-used, those data might remain and make their way into a new packet. The evaluator shall check to ensure that the TSS describes packet processing to the extent that they can determine that no data will be reused when processing network packets. The evaluator shall ensure that this description at a minimum describes how the previous data are zeroized/overwritten, and at what point in the buffer processing this occurs.

6.1.4 Identification and Authentication (FIA)

6.1.4.1 FIA_AFL.1 Authentication Failure Handling

FIA_AFL.1.1 The TSF shall detect when an Administrator configurable positive integer of successive unsuccessful authentication attempts occur related to administrators attempting to authenticate remotely.

FIA_AFL.1.2 When the defined number of unsuccessful authentication attempts has been met, the TSF shall prevent the offending remote administrator from successfully authenticating until **account unlock action** is taken by a local Administrator.

EP Application Note: *This requirement does not apply to an administrator at the local console, since it does not make sense to lock a local administrator's account in this fashion. This could be addressed by (for example) requiring a separate account for local administrators or having the authentication mechanism implementation distinguish local and remote login attempts. The "action" taken by a local administrator is implementation specific and would be defined in the administrator guidance (for example, lockout reset or password reset). The ST author chooses one of the selections for handling of*

authentication failures depending on how the TOE has implemented this handler.

Assurance Activity:

TSS: The evaluator shall examine the TSS to determine that it contains a description, for each supported method for remote administrative actions, of how successive unsuccessful authentication attempts are detected and tracked. The TSS shall also describe the method by which the remote administrator is prevented from successfully logging on to the TOE, and the actions necessary to restore this ability.

Guidance: The evaluator shall also examine the operational guidance to ensure that instructions for configuring the number of successive unsuccessful authentication attempts (1.1) and time period (1.2, if implemented) are provided, and that the process of allowing the remote administrator to once again successfully log on is described for each "action" specified (if that option is chosen). If different actions or mechanisms are implemented depending on the secure protocol employed (e.g., TLS vs. SSH), all must be described.

Test: The evaluator shall perform the following tests for IPsec, and for each other method by which remote administrators access the TOE (e.g., TLS, SSH):

- Test 1: The evaluator shall use the operational guidance to configure the number of successive unsuccessful authentication attempts allowed by the TOE. The evaluator shall test that once the limit is reached, attempts with valid credentials are not successful. For each action specified by the requirement, the evaluator shall show that following the operational guidance and performing each action to allow the remote administrator access are successful.
- Test 2: The evaluator shall use the operational guidance to configure the number of successive unsuccessful authentication attempts allowed by the TOE and a time period after which valid logins will be allowed for a remote administrator. After exceeding the specified number of invalid login attempts and showing that valid login is not possible, the evaluator shall show that waiting for the interval defined by the time period before another access attempt will result in the ability for the remote administrator to successfully log on using valid credentials.

6.1.4.2 FIA_PMG_EXT.1 Password Management

FIA_PMG_EXT.1.1 The TSF shall provide the following password management capabilities for administrative passwords:

1. Passwords shall be able to be composed of any combination of upper and lower case letters, numbers, and the following special characters: "!", "@", "$", "%", "^", "&", "*", "(", ")", "", "#", "", "+", ",", "-", ".", "/", ":", ";", "<", "=", ">", "?", "[", "\", "]", "_", "`", "{", "|", "}", "~";

2. Minimum password length shall settable by the Security Administrator, and support passwords of 15 characters or greater;

PP Application Note: *The ST author selects the special characters that are supported by TOE; they may optionally list additional special characters supported using the assignment. "Administrative passwords" refers to passwords used by administrators at the local console or over protocols that support passwords, such as SSH and HTTPS.*

Assurance Activity: The evaluator shall examine the operational guidance to determine that it provides guidance to security administrators on the composition of strong passwords, and that it provides instructions on setting the minimum password length. The evaluator shall also perform the following tests. Note that one or more of these tests can be performed with a single test case.

- Test 1: The evaluator shall compose passwords that either meet the requirements, or fail to meet the requirements, in some way. For each password, the evaluator shall verify that the TOE supports the password. While the evaluator is not required (nor is it feasible) to test all possible compositions of passwords, the evaluator shall ensure that all characters, rule characteristics, and a minimum length listed in the requirement are supported, and justify the subset of those characters chosen for testing.

6.1.4.3 FIA_UIA_EXT.1 User Identification and Authentication

FIA_UIA_EXT.1.1 The TSF shall allow the following actions prior to requiring the non-TOE entity to initiate the identification and authentication process:

- Display the warning banner in accordance with FTA_TAB.1;
- **Establish SSH Session (Management Interface);**
- **OSPF (Protected Network Interface);**
- **Initiate IKEv2 (Public Network Interface).**

FIA_UIA_EXT.1.2 The TSF shall require each administrative user to be successfully identified and authenticated before allowing any other TSF-mediated actions on behalf of that administrative user.

PP Application Note: *This requirement applies to users (administrators and external IT entities) of services available from the TOE directly, and not services available by connecting through the TOE. While it should be the case that few or no services are available to external entities prior to identification and authentication, if there are some available (perhaps ICMP echo) these should be listed in the assignment statement; otherwise "no other actions" should be selected.*

Authentication can be password-based through the local console or through a protocol that supports passwords (such as SSH), or be certificate based (SSH, TLS).

For communications with external IT entities (e.g., an audit server or NTP server, for instance), such connections must be performed in accordance with FTP_ITC.1, whose protocols perform identification and authentication. This means that such communications (e.g., establishing the IPsec connection to the authentication server) would not have to be specified in the assignment, since establishing the connection "counts" as initiating the identification and authentication process.

Assurance Activity: The evaluator shall examine the TSS to determine that it describes the logon process for each logon method (local, remote (HTTPS, SSH, etc.)) supported for the product. This description shall contain information pertaining to the credentials allowed/used, any protocol transactions that take place, and what constitutes a "successful logon". The evaluator shall examine the operational guidance to determine that any necessary preparatory steps (e.g., establishing credential material such as pre-shared keys, tunnels, certificates, etc.) to logging in are described. For each supported the login method, the evaluator shall ensure the operational guidance provides clear instructions for successfully logging on. If configuration is necessary to ensure the services provided before login are limited, the evaluator shall determine that the operational guidance provides sufficient instruction on limiting the allowed services.

The evaluator shall perform the following tests for each method by which administrators access the TOE (local and remote), as well as for each type of credential supported by the login method:

- Test 1: The evaluator shall use the operational guidance to configure the appropriate credential supported for the login method. For that credential/login method, the evaluator shall show that providing correct I&A information results in the ability to access the system, while providing incorrect information results in denial of access.
- Test 2: The evaluator shall configure the services allowed (if any) according to the operational guidance, and then determine the services available to an external remote entity. The evaluator shall determine that the list of services available is limited to those specified in the requirement.
- Test 3: For local access, the evaluator shall determine what services are available to a local administrator prior to logging in, and make sure this list is consistent with the requirement.

6.1.4.4 FIA_UAU_EXT.2 Password-based Authentication Mechanism

FIA_UAU_EXT.2.1 The TSF shall provide a local password-based authentication mechanism, **SSH Public Key based authentication** to perform administrative user authentication.

Assurance Activity: Assurance activities for this requirement are covered under those for FIA_UIA_EXT.1. If other authentication mechanisms are specified, the evaluator shall include those methods in the activities for FIA_UIA_EXT.1.

6.1.4.5 FIA_UAU.7 Protected Authentication Feedback

FIA_UAU.7.1 The TSF shall provide only obscured feedback to the administrative user while the authentication is in progress at the local console.

PP Application Note: *"Obscured feedback" implies the TSF does not produce a visible display of any authentication data entered by a user (such as the echoing of a password), although an obscured indication of progress may be provided (such as an asterisk for each character). It also implies that the TSF does not return any information during the authentication process to the user that may provide any indication of the authentication data.*

Assurance Activity: The evaluator shall perform the following test for each method of local login allowed:

- Test 1: The evaluator shall locally authenticate to the TOE. While making this attempt, the evaluator shall verify that at most obscured feedback is provided while entering the authentication information.

6.1.4.6 FIA_X509_EXT.1 Extended: X.509 Certificates

FIA_X509_EXT.1.1 The TSF shall use X.509v3 certificates as defined by RFC 5280 to support authentication for IPsec and TLS connections.

FIA_X509_EXT.1.2 The TSF shall store and protect certificate(s) from unauthorized deletion and modification.

FIA_X509_EXT.1.3 The TSF shall provide the capability for authenticated Administrators to load X.509v3 certificates into the TOE for use by the security functions specified in this ***ST ~~PP~~***.

FIA_X509_EXT.1.4 The TSF shall generate a Certificate Request Message as specified in RFC 2986 and be able to provide the following information in the request: public key, Common Name, Organization, Organizational Unit, and Country.

EP Application Note: *The public key referenced in FIA_X509_EXT.1.4 is the public key portion of the public- private key pair generated by the TOE as specified in FCS_CKM.1(2).*

FIA_X509_EXT.1.5 The TSF shall validate the certificate using a Certificate Revocation List (CRL) as specified in RFC 5759.

EP Application Note: *While the choice of revocation method employed is left to the ST author, future versions of the EP will mandate both methods be available to the TOE's Administrator.*

FIA_X509_EXT.1.6 The TSF shall validate a certificate path by ensuring the presence of the basicConstraints extension is present and the cA flag is set to TRUE for all CA certificates.

FIA_X509_EXT.1.7 The TSF shall not treat a certificate as a CA certificate if the basicConstraints extension is not present or the cA flag is not set to TRUE.

FIA_X509_EXT.1.8 The TSF shall not establish an SA if a certificate or certificate path is deemed invalid.

FIA_X509_EXT.1.9 The TSF shall not establish an SA if the distinguished name (DN) contained in a certificate does not match the expected DN for the entity attempting to establish a connection.

FIA_X509_EXT.1.10 When the TSF cannot establish a connection to determine the validity of a certificate, the TSF shall, at the option of the administrator, establish an SA or disallow the establishment of an SA.

EP Application Note: *The intent of FIA_X509_EXT.1.**10~~8~~** is that the TOE is configurable to allow or disallow session establishment if the TOE cannot connect to an entity responsible for providing certificate validation information. For instance, if a CRL cannot be obtained because a machine is down, or the network path is broken, the administrator may elect to configure the TOE to allow sessions to continue to be established, rather than terminate the TOE's ability to establish any new SAs because it cannot reach the CA.*

Assurance Activity:

TSS: The TSS shall describe all certificate stores implemented that contain certificates used to meet the requirements of this EP. This description shall contain information pertaining to how certificates are loaded into the store, and how the store is protected from unauthorized access. The TSS description will also include a discussion as to how the TOE forms a certification path as specified in the standard and how certificates are validated (CRL and/or OCSP are included in the discussion, as well as the certificate path validation algorithm).

Guidance: The evaluator shall verify that the operational guidance describes how to the administrator loads certificates into the certificate store. If the level of protection can managed by the administrator, the guidance provides a description of how to manage the protection mechanism. The guidance instructs the administrator how to generate a key pair and how to generate a Certificate Request Message to the CA.

The guidance documentation provides instructions how to select the method used for checking, as well as how to setup a protected communication path with the entity providing the information pertaining to certificate validity.

How the administrator can configure the TOE to either allow or disallow the establishment of an SA is also described in the operational guidance.

Test: The tests associated with this component are bundled with the FCS_IPSEC_EXT.1.12 requirements.

6.1.5 Security Management (FMT)

6.1.5.1 FMT_MOF.1 Management of Security Functions Behavior

FMT_MOF.1.1 The TSF shall restrict the ability to enable, disable, determine and modify the behavior of all of the security functions of the TOE identified in this EP to an authenticated Administrator.

6.1.5.2 FMT_MTD.1 Management of TSF Data (for general TSF data)

FMT_MTD.1.1 The TSF shall restrict the ability to manage the TSF data to the Security Administrators.

PP Application Note: *The word "manage" includes but is not limited to create, initialize, view, change default, modify, delete, clear, and append. This requirement is intended to be the "default" requirement for management of TSF data; other iterations of FMT_MTD should place different restrictions or operations available on the specifically-identified TSF data. TSF data includes cryptographic information as well; managing these data would include the association of a cryptographic protocol with an interface, for instance.*

Assurance Activity: The evaluator shall review the operational guidance to determine that each of the TSF-data-manipulating functions implemented in response to the requirements of the NDPP is identified, and that configuration information is provided to ensure that only administrators have access to the functions. The evaluator shall examine the TSS to determine that, for each administrative function identified in the operational guidance, those that are accessible through an interface prior to administrator log-in are identified. For each of these functions, the evaluator shall also confirm that the TSS details how the ability to manipulate the TSF data through these interfaces is disallowed for non-administrative users.

6.1.5.3 FMT_SMF.1 Specification of Management Functions

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

- Ability to configure the cryptographic functionality,
- Ability to configure the IPsec functionality,
- Ability to enable, disable, determine and modify the behavior of all the security functions of the TOE identified in this ~~***EP***~~ ***ST*** to the Administrator,
- Ability to configure all security management functions identified in other sections of this ~~***EP***~~ ***ST***.

PP Application Note: *The TOE must provide functionality for both local and remote administration, as well as the capability for the administrator to verify that updates received came from a trusted source. They must be capable of performing this action using digital signatures, and optionally a published hash. The ST author chooses whether the published hash verification option is available using the first selection, which must match the corresponding selection in FPT_TUD_EXT.1.3. If the TOE offers the ability for the administrator to configure the services available prior to identification or authentication, or if any of the cryptographic functionality on the TOE can be configured, then the ST author makes the appropriate choice or choices in the second selection, otherwise select "no other capabilities."*

EP Assurance Activity:

TSS: The evaluator shall verify that the TSS describes how the Packet filter firewall rules can be configured. Note that this activity should have been addressed with the TSS assurance activities for FPF_RUL_EXT.1.

Guidance: The evaluator shall verify that the operational guidance describes how to configure the Packet filter firewall rules, including how to set any configurable defaults and how to configure each of the applicable rule attributes, actions, and associated interfaces. The evaluator must ensure that the operational guidance also provides instruction that would allow an administrator to ensure that configured rules are properly ordered. Note that this activity should have been addressed with the Guidance assurance activities for FPF_RUL_EXT.1.

Test:

- Test 1: The evaluator shall devise tests that demonstrate that the functions used to configure the Packet filter firewall rules yield expected changes in the rules that they are correctly enforced. A number of rule combination and ordering scenarios need to be configured and tested by attempting to pass both valid and invalid network traffic through the TOE. Note that this activity should have been addressed with a combination of the Test assurance activities for FPF_RUL_EXT.1.

Assurance Activity: The security management functions for FMT_SMF.1 are distributed throughout the PP and are included as part of the requirements in FMT_MTD, FPT_TST_EXT, and any cryptographic management functions specified in the reference standards. Compliance to these requirements satisfies compliance with FMT_SMF.1.

6.1.5.4 FMT_SMR.2 Restrictions on Security Roles

FMT_SMR.2.1 The TSF shall maintain the roles:

- Authorized Administrator

FMT_SMR.2.2 The TSF shall be able to associate users with roles.

FMT_SMR.2.3 The TSF shall ensure that the conditions

- Authorized Administrator role shall be able to administer the TOE locally;
- Authorized Administrator role shall be able to administer the TOE remotely;

are satisfied.

PP Application Note: *FMT_SMR.2.2 requires that user accounts be associated with only one role. However, note that multiple users may have the same role, and the TOE is not required to restrict roles to a single person.*

FMT_SMR.2.3 requires that an authorized administrator be able to administer the TOE through the local console and through a remote mechanism (IPsec, SSH, TLS, TLS/HTTPS). For multiple component TOEs, only

the TOE components providing the management control and configuration of the other TOE components require a local administration interface.

Assurance Activity: The evaluator shall review the operational guidance to ensure that it contains instructions for administering the TOE both locally and remotely, including any configuration that needs to be performed on the client for remote administration. In the course of performing the testing activities for the evaluation, the evaluator shall use all supported interfaces, although it is not necessary to repeat each test involving an administrative action with each interface. The evaluator shall ensure, however, that each supported method of administering the TOE that conforms to the requirements of the NDPP be tested; for instance, if the TOE can be administered through a local hardware interface; SSH; and TLS/HTTPS; then all three methods of administration must be exercised during the evaluation team's test activities.

6.1.6 Packet Filtering (FPF)

6.1.6.1 FPF_RUL_EXT.1 Packet Filtering

FPF_RUL_EXT.1.1 The TSF shall perform Packet Filtering on network packets processed by the TOE.

Assurance Activity:

TSS: The evaluator shall verify that the TSS provide a description of the TOE's initialization/startup process, which clearly indicates where processing of network packets begins to take place, and provides a discussion that supports the assertion that packets cannot flow during this process. The evaluator shall verify that the TSS also includes a narrative that identifies the components (e.g., active entity such as a process or task) involved in processing the network packets and describes the safeguards that would prevent packets flowing through the TOE without applying the ruleset in the event of a component failure. This could include the failure of a component, such as a process being terminated, or a failure within a component, such as memory buffers full and cannot process packets.

Guidance: The operational guidance associated with this requirement is assessed in the subsequent test assurance activities.

Tests:

- Test 1: The evaluator shall attempt to get network traffic to flow through the TOE while the TOE is being initialized. A steady flow of network packets that would otherwise be denied by the ruleset should be directed at the TOE's interfaces, with packet sniffers listening to see if any network traffic is allowed through.

Note: The remaining testing associated with application of the ruleset is addressed in the subsequent test assurance activities.

FPF_RUL_EXT.1.2 The TSF shall process the following network traffic protocols:

- Internet Protocol (IPv4)

- ~~***Internet Protocol version 6 (IPv6)***~~
- Transmission Control Protocol (TCP)
- User Datagram Protocol (UDP)

and be capable of inspecting network packet header fields defined by the following RFCs to the extent mandated in the other elements of this SFR

- RFC 791 (IPv4)
- ~~***RFC 2460 (IPv6)***~~
- RFC 793 (TCP)
- RFC 768 (UDP).

TOE Application Note: *Refinement: IPv6 was deleted, because the VPNGW TRRT indicated that only IPv4 or IPv6 must be supported.*

EP Application Note: *This element identifies the protocols and references the protocol definitions that serve to define to what extent the network traffic can be interpreted by the TOE when importing (receiving network traffic or ingress) and exporting (sending – or forming to be sent - network traffic or egress).*

While the protocol formatting specified in the RFCs is still used, many RFCs define behaviors which are no longer considered safe to follow. For example, RFC792 defined the "Redirect" ICMP type, which is not considered safe to honor when it might come from an adversary; the "source quench" message, which is insecure because its source cannot be validated.

Assurance Activity:

TSS: The evaluator shall verify that the TSS indicates that the following protocols are supported:

- RFC 791 (IPv4)
- ~~***RFC 2460 (IPv6)***~~
- RFC 793 (TCP)
- RFC 768 (UDP)

The evaluator shall verify that the TSS describes how conformance with the identified RFCs has been determined by the TOE developer (e.g., third party interoperability testing, protocol compliance testing).

Guidance: The evaluator shall verify that the operational guidance indicates that the following protocols are supported:

- RFC 791 (IPv4)
- ~~***RFC 2560 (IPv6)***~~
- RFC 793 (TCP)
- RFC 768 (UDP)

The guidance will describe the other protocols contained within the ST (e.g., IPsec, IKE, potentially HTTPS, SSH, and TLS) that are processed by the TOE. The evaluator ensures it is made clear what protocols were not considered as part of the TOE evaluation.

Tests: The testing associated with this requirement is addressed in the subsequent test assurance activities.

FPF_RUL_EXT.1.3 The TSF shall allow the definition of Packet Filtering rules using the following network protocol fields:

- IPv4
  - Source address
  - Destination Address
  - Protocol
- ~~***IPv6***~~
  - ~~***Source address***~~
  - ~~***Destination Address***~~
  - ~~***Next Header (Protocol)***~~
- TCP
  - Source Port
  - Destination Port
- UDP
  - Source Port
  - Destination Port

and distinct interface.

TOE Application Note: *Refinement: IPv6 was deleted, because the VPNGW TRRT indicated that only IPv4 or IPv6 must be supported.*

EP Application Note: *This element identifies the various attributes that are applicable when constructing rules to be enforced by this requirement – the applicable interface is a property of the TOE and the rest of the identified attributes are defined in the associated RFCs. Note that the Protocol is the IPv4 field (in IPv6 this field is called the "next header" that identifies the applicable protocol, such as TCP, UDP, ICMP, etc.. Also, 'Interface' identified above is the external port where the applicable network traffic was received or alternately will be sent.*

FPF_RUL_EXT.1.4 The TSF shall allow the following operations to be associated with Packet Traffic Filtering rules: permit, deny, and log.

EP Application Note: *This element defines the operations that can be associated with rules used to match network traffic. Note that the data to be logged is identified in the Security Audit requirements, see Section 6.1.1.*

FPF_RUL_EXT.1.5 The TSF shall allow the Packet Traffic Filtering rules to be assigned to each distinct network interface.

EP Application Note: *This element identifies where rules can be assigned. Specifically, a conforming TOE must be able to assign filtering rules specific to each of its available and identifiable distinct network interfaces that handle layer 3 and 4 network traffic. Identifiable means the interface is unique and identifiable within the TOE, and does not necessarily require the interface to be visible from the network perspective (e.g., does not need to have an IP address assigned to it). A distinct network interface is one or more physical connections that share a common logical path into the TOE. For example, the TOE might have a small form-factor pluggable (SFP) port supporting SFP modules that expose a number of physical network ports, but since a common driver is used for all external ports they can be treated as a single distinct network interface.*

Note that there could be a separate ruleset for each interface or alternately a shared ruleset that somehow associates rules with specific interfaces.

Assurance Activity:

TSS: The evaluator shall verify that the TSS describes a Packet Filtering policy and the following attributes are:

- IPv4
  - Source address
  - Destination Address
  - Protocol
- ~~***IPv6***~~
  - ~~***Source address***~~
  - ~~***Destination Address***~~
  - ~~***Next Header (Protocol)***~~
- TCP
  - Source Port
  - Destination Port
- UDP
  - Source Port
  - Destination Port

The evaluator shall verify that each rule can identify the following actions: permit, deny, and log.

The evaluator shall verify that the TSS identifies all interface types subject to the Packet Filtering policy and explains how rules are associated with distinct network interfaces. Where interfaces can be grouped into a common interface type (e.g., where the same internal logical path is used,

perhaps where a common device driver is used) they can be treated collectively as a distinct network interface.

Guidance: The evaluators shall verify that the operational guidance identifies the following attributes as being configurable within Packet filtering rules for the associated protocols:

- IPv4
  - Source address
  - Destination Address
  - Protocol
- ~~***IPv6***~~
  - ~~***Source address***~~
  - ~~***Destination Address***~~
  - ~~***Next Header (Protocol)***~~
- TCP
  - Source Port
  - Destination Port
- UDP
  - Source Port
  - Destination Port

The evaluator shall verify that the operational guidance indicates that each rule can identify the following actions: permit, deny, and log.

The evaluator shall verify that the operational guidance explains how rules are associated with distinct network interfaces.

The evaluator shall verify that the operational guidance explains how to determine the interface type of a distinct network interface (e.g., how to determine the device driver for a distinct network interface).

Tests:

- Test 1: The evaluator shall use the instructions in the operational guidance to test that packet filter rules can be created that permit, deny, and log packets for each of the following attributes:
  - IPv4
    - Source address
    - Destination Address
    - Protocol
  - ~~***IPv6***~~
    - ~~***Source address***~~

- ~~***Destination Address***~~
- ~~***Protocol***~~

o TCP

- Source Port
- Destination Port

o UDP

- Source Port
- Destination Port

- Test 2: Repeat the test assurance activity above to ensure that Packet filtering rules can be defined for each distinct network interface type supported by the TOE.

Note that these test activities should be performed in conjunction with those of FPF_RUL_EXT.1.7 where the effectiveness of the rules is tested; here the evaluator is just ensuring the guidance is sufficient and the TOE supports the administrator creating a ruleset based on the above attributes. The test activities for FPF_RUL_EXT.1.7 define the protocol/attribute combinations required to be tested. If those combinations are configured manually, that will fulfill the objective of these test activities, but if those combinations are configured otherwise (e.g., using automation), these test activities may be necessary in order to ensure the guidance is correct and the full range of configurations can be achieved by a TOE administrator.

FPF_RUL_EXT.1.6 The TSF shall process the applicable Packet Filtering rules (as determined in accordance with FPF_RUL_EXT.1.5) in the following order: Administrator-defined.

EP Application Note: *This element requires that an administrator is able to define the order in which configured filtering rules are processed for matches.*

Assurance Activity:

TSS: The evaluator shall verify that the TSS describes the algorithm applied to incoming packets, including the processing of default rules, determination of whether a packet is part of an established session, and application of administrator defined and ordered ruleset.

Guidance: The evaluator shall verify that the operational guidance describes how the order of Packet filtering rules is determined and provides the necessary instructions so that an administrator can configure the order of rule processing.

Tests:

- Test 1: The evaluator shall devise two equal Packet filtering rules with alternate operations – permit and deny. The rules should then be deployed in two distinct orders and in each case the evaluator shall ensure that the first rule is enforced in both cases by

generating applicable packets and using packet capture and logs for confirmation.

- Test 2: The evaluator shall repeat the procedure above, except that the two rules should be devised where one is a subset of the other (e.g., a specific address vs. a network segment). Again, the evaluator should test both orders to ensure that the first is enforced regardless of the specificity of the rule.

FPF_RUL_EXT.1.7 The TSF shall deny packet flow if a matching rule is not identified.

EP Application Note: *This element requires that the behavior is always to deny network traffic when no rules apply.*

Assurance Activity:

TSS: The evaluator shall verify that the TSS describes the process for applying Packet filtering rules and also that the behavior (either by default, or as configured by the administrator) is to deny packets when there is no rule match unless another required conditions allows the network traffic (i.e., FPF_RUL_EXT.1.6 or FPF_RUL_EXT.1.7).

Guidance: The evaluator shall verify that the operational guidance describes the behavior if no rules or special conditions apply to the network traffic. If the behavior is configurable, the evaluator shall verify that the operational guidance provides the appropriate instructions to configure the behavior to deny packets with no matching rules.

Tests:

- Test 1: The evaluator shall configure the TOE to permit and log each defined IPv4 Transport Layer Protocol (see table 9-1 Defined Protocol-specific Values) in conjunction with a specific source address and specific destination address, specific source address and wildcard destination address, wildcard source address and specific destination address, and wildcard source address and wildcard destination address. The evaluator shall generate packets matching each defined IPv4 Transport Layer Protocol and within the configured source and destination addresses in order to ensure that they are permitted (i.e., by capturing the packets after passing through the TOE) and logged.
- Test 2: The evaluator shall configure the TOE to permit all traffic except to deny and log each defined IPv4 Transport Layer Protocol (see table 9-1 Defined Protocol-specific Values) in conjunction with a specific source address and specific destination address, specific source address and wildcard destination address, wildcard source address and specific destination address, and wildcard source address and wildcard destination address. The evaluator shall generate packets matching each defined IPv4 Transport Layer Protocol and within the configured source and destination addresses in order to ensure that they are denied (i.e., by capturing no applicable packets passing through the TOE) and logged.

- Test 3: The evaluator shall configure the TOE to permit and log each defined IPv4 Transport Layer Protocol (see table 9-1 Defined Protocol-specific Values) in conjunction with a specific source address and specific destination address, specific source address and wildcard destination address, wildcard source address and specific destination address, and wildcard source address and wildcard destination address. Additionally, the evaluator shall configure the TOE to deny and log each defined IPv4 Transport Layer Protocol (see table 9-1 Defined Protocol-specific Values) in conjunction with different (than those permitted above) combinations of a specific source address and specific destination address, specific source address and wildcard destination address, wildcard source address and specific destination address, and wildcard source address and wildcard destination address. The evaluator shall generate packets matching each defined IPv4 Transport Layer Protocol and outside the scope of all source and destination addresses configured above in order to ensure that they are denied (i.e., by capturing no applicable packets passing through the TOE).

- ***~~Test 4: The evaluator shall configure the TOE to permit and log each defined IPv6 Transport Layer Protocol (see table 9-1 Defined Protocol-specific Values) in conjunction with a specific source address and specific destination address, specific source address and wildcard destination address, wildcard source address and specific destination address, and wildcard source address and wildcard destination address. The evaluator shall generate packets matching each defined IPv6 Transport Layer Protocol and within the configured source and destination addresses in order to ensure that they are permitted (i.e., by capturing the packets after passing through the TOE) and logged.~~***

- ***~~Test 5: The evaluator shall configure the TOE to permit all traffic except to deny and log each defined IPv6 Transport Layer Protocol (see table 9-1 Defined Protocol-specific Values) in conjunction with a specific source address and specific destination address, specific source address and wildcard destination address, wildcard source address and specific destination address, and wildcard source address and wildcard destination address. The evaluator shall generate packets matching each defined IPv6 Transport Layer Protocol and within the configured source and destination addresses in order to ensure that they are denied (i.e., by capturing no applicable packets passing through the TOE) and logged.~~***

- ***~~Test 6: The evaluator shall configure the TOE to permit and log each defined IPv6 Transport Layer Protocol (see table 9-1 Defined Protocol-specific Values) in conjunction with a specific source address and specific destination address, specific source address~~***

~~and wildcard destination address, wildcard source address and specific destination address, and wildcard source address and wildcard destination address. Additionally, the evaluator shall configure the TOE to deny and log each defined IPv6 Transport Layer Protocol (see table 9-1 Defined Protocol-specific Values) in conjunction with different (than those permitted above) combinations of a specific source address and specific destination address, specific source address and wildcard destination address, wildcard source address and specific destination address, and wildcard source address and wildcard destination address. The evaluator shall generate packets matching each defined IPv6 Transport Layer Protocol and outside the scope of all source and destination addresses configured above in order to ensure that they are denied (i.e., by capturing no applicable packets passing through the TOE).~~

- Test 7: The evaluator shall configure the TOE to permit and log protocol 6 (TCP) using a selected source port, a selected destination port, and a selected source and destination port combination. The evaluator shall generate packets matching the configured source and destination TCP ports in order to ensure that they are permitted (i.e., by capturing the packets after passing through the TOE) and logged.
- Test 8: The evaluator shall configure the TOE to deny and log protocol 6 (TCP) using a selected source port, a selected destination port, and a selected source and destination port combination. The evaluator shall generate packets matching the configured source and destination TCP ports in order to ensure that they are denied (i.e., by capturing no applicable packets passing through the TOE) and logged.
- Test 9: The evaluator shall configure the TOE to permit and log protocol 17 (UDP) using a selected source port, a selected destination port, and a selected source and destination port combination. The evaluator shall generate packets matching the configured source and destination UDP ports in order to ensure that they are permitted (i.e., by capturing the packets after passing through the TOE) and logged. Here the evaluator ensures that the UDP port 500 (IKE) is included in the set of tests.
- Test 10: The evaluator shall configure the TOE to deny and log protocol 17 (UDP) using a selected source port, a selected destination port, and a selected source and destination port combination. The evaluator shall generate packets matching the configured source and destination UDP ports in order to ensure that they are denied (i.e., by capturing no applicable packets passing through the TOE) and logged. Again, the evaluator ensures that UDP port 500 is included in the set of tests.

6.1.7 Protection of the TSF (FPT)

6.1.7.1 FPT_SKP_EXT.1 Protection of TSF Data (for reading of all symmetric keys)

FPT_SKP_EXT.1.1 The TSF shall prevent reading of all pre-shared keys, symmetric keys, and private keys.

PP Application Note: *The intent of the requirement is that an administrator is unable to read or view the identified keys (stored or ephemeral) through "normal" interfaces. While it is understood that the administrator could directly read memory to view these keys, do so is not a trivial task and may require substantial work on the part of an administrator. Since the administrator is considered a trusted agent, it is assumed they would not endeavor in such an activity.*

Assurance Activity: The evaluator shall examine the TSS to determine that it details how any pre-shared keys, symmetric keys, and private keys are stored and that they are unable to be viewed through an interface designed specifically for that purpose, as outlined in the application note. If these values are not stored in plaintext, the TSS shall describe how they are protected/obscured.

6.1.7.2 FPT_APW_EXT.1 Protection of Administrator Passwords

FPT_APW_EXT.1.1 The TSF shall store passwords in non-plaintext form.

FPT_APW_EXT.1.2 The TSF shall prevent the reading of plaintext passwords.

PP Application Note: *The intent of the requirement is that raw password authentication data are not stored in the clear, and that no user or administrator is able to read the plaintext password through "normal" interfaces. An all-powerful administrator of course could directly read memory to capture a password but is trusted not to do so.*

In this version of the PP there are no requirements on the method used to store the passwords in non-plaintext form, but cryptographic methods based on the requirements in FCS_COP are preferred. In future versions of the NDPP, FCS_COP-based cryptographic methods that conform to the Level 2 Credential Storage requirements from NIST SP 800-63 will be required.

Assurance Activity: The evaluator shall examine the TSS to determine that it details all authentication data that are subject to this requirement, and the method used to obscure the plaintext password data when stored. The TSS shall also detail passwords are stored in such a way that they are unable to be viewed through an interface designed specifically for that purpose, as outlined in the application note.

6.1.7.3 FPT_FLS.1 Fail Secure

FPT_FLS.1.1 The TSF shall shutdown when the following types of failures occur: failure of the power-on self-tests, failure of integrity check of the TSF executable image, failure of noise source health tests.

EP Application Note: *The failures relevant to this requirement are the FPT_TST_EXT.1.1 requirement in the NDPP, and the FPT_TST_EXT.1.2 requirement specified in the EP.*

Assurance Activity:

TSS: The evaluator shall ensure the TSS describes how the TOE ensures a shutdown upon a self-test failure, a failed integrity check of the TSF executable image, or a failed health test of the noise source. If there are instances when a shut-down does not occur, e.g., a failure is deemed non-security relevant, those cases are identified and a rationale supporting the classification and justification why the TOE's ability to enforce its security policies is not affected.

6.1.7.4 FPT_STM.1 Reliable Time Stamps

FPT_STM.1.1 The TSF shall be able to provide reliable time stamps for its own use.

Assurance Activity: The evaluator shall examine the TSS to ensure that it lists each security function that makes use of time. The TSS provides a description of how the time is maintained and considered reliable in the context of each of the time related functions.

The evaluator examines the operational guidance to ensure it instructs the administrator how to set the time. If the TOE supports the use of an NTP server, the operational guidance instructs how a communication path is established between the TOE and the NTP server, and any configuration of the NTP client on the TOE to support this communication.

- Test 1: The evaluator uses the operational guide to set the time. The evaluator shall then use an available interface to observe that the time was set correctly.
- Test2: [conditional] If the TOE supports the use of an NTP server; the evaluator shall use the operational guidance to configure the NTP client on the TOE, and set up a communication path with the NTP server. The evaluator will observe that the NTP server has set the time to what is expected. If the TOE supports multiple protocols for establishing a connection with the NTP server, the evaluator shall perform this test using each supported protocol claimed in the operational guidance.

6.1.7.5 FPT_TUD_EXT.1 Trusted Update

FPT_TUD_EXT.1.1 The TSF shall provide security administrators the ability to query the current version of the TOE firmware/software.

FPT_TUD_EXT.1.2 The TSF shall provide security administrators the ability to initiate updates to TOE firmware/software.

FPT_TUD_EXT.1.3 The TSF 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.

EP Application Note: *The NDPP provides an option of which method of verification the ST Author wishes to specify. For compliance with the EP, a digital signature mechanism (one of those specified in FCS_COP.1(2) must be employed.*

PP Application Note: *The digital signature mechanism referenced in the third element is the one specified in FCS_COP.1(2). The published hash referenced is generated by one of the functions specified in FCS_COP.1(3). The ST author should choose the mechanism implemented by the TOE; it is acceptable to implement both mechanisms.*

Assurance Activity: Updates to the TOE either have a hash associated with them, or are signed by an authorized source. If digital signatures are used, the definition of an authorized source is contained in the TSS, along with a description of how the certificates used by the update verification mechanism are contained on the device. The evaluator ensures this information is contained in the TSS. The evaluator also ensures that the TSS (or the operational guidance) describes how the candidate updates are obtained; the processing associated with verifying the digital signature or calculating the hash of the updates; and the actions that take place for successful (hash or signature was verified) and unsuccessful (hash or signature could not be verified) cases. The evaluator shall perform the following tests:

- Test 1: The evaluator performs the version verification activity to determine the current version of the product. The evaluator obtains a legitimate update using procedures described in the operational guidance and verifies that it is successfully installed on the TOE. Then, the evaluator performs a subset of other assurance activity tests to demonstrate that the update functions as expected. After the update, the evaluator performs the version verification activity again to verify the version correctly corresponds to that of the update.
- Test 2: The evaluator performs the version verification activity to determine the current version of the product. The evaluator obtains or produces an illegitimate update, and attempts to install it on the TOE. The evaluator verifies that the TOE rejects the update.

6.1.7.6 FPT_TST_EXT.1 TSF Testing

FPT_TST_EXT.1.1 The TSF 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 TSF shall provide the capability to verify the integrity of stored TSF executable code when it is loaded for execution through the use of the TSF-provided cryptographic service specified in FCS_COP.1(2).

EP Application Note: *The NDPP contains one element for this component, which simply requires a suite of self-tests to demonstrate correct operation of the TSF. This element is added to that component to comply with the EP.*

Assurance Activity: The evaluator shall examine the TSS to ensure that it details the self tests that are run by the TSF on start-up; this description should include an outline of what the tests are actually doing (e.g., rather than saying "memory is tested", a description similar to "memory is tested by writing a value to each memory location and reading it back to ensure it is identical to what was written" shall be used). The evaluator shall ensure that the TSS

makes an argument that the tests are sufficient to demonstrate that the TSF is operating correctly.

The evaluator shall also ensure that the operational guidance describes the possible errors that may result from such tests, and actions the administrator should take in response; these possible errors shall correspond to those described in the TSS.

6.1.8 TOE Access (FTA)

6.1.8.1 FTA_SSL_EXT.1 TSF-initiated Session Locking

FTA_SSL_EXT.1.1 The TSF shall, for local interactive sessions,

- terminate the session

after a Security Administrator-specified time period of inactivity.

Assurance Activity: The evaluator shall perform the following test:

- Test 1: The evaluator follows the operational guidance to configure several different values for the inactivity time period referenced in the component. For each period configured, the evaluator establishes a local interactive session with the TOE. The evaluator then observes that the session is either locked or terminated after the configured time period. If locking was selected from the component, the evaluator then ensures that re-authentication is needed when trying to unlock the session.

6.1.8.2 FTA_SSL.3 TSF-initiated Termination

FTA_SSL.3.1 The TSF shall terminate a remote interactive session after a Security Administrator-configurable time interval of session inactivity.

Assurance Activity: The evaluator shall perform the following test:

- Test 1: The evaluator follows the operational guidance to configure several different values for the inactivity time period referenced in the component. For each period configured, the evaluator establishes a remote interactive session with the TOE. The evaluator then observes that the session is terminated after the configured time period.

6.1.8.3 FTA_SSL.3(2) TSF-initiated Termination

FTA_SSL.3.1(2) The TSF shall terminate a remote VPN client session after a Administrator-configurable time interval of session inactivity.

EP Application Note: *This requirement exists in the NDPP, however it is intended to address a remote administrative interactive session. Here, the requirement applies to a VPN client that has established a SA. After some configurable time period without any activity, the connection between the VPN headend and client is terminated. If the ST author is including the requirements for a VPN headend in their ST, this requirement should be iterated along with the requirement in the NDPP.*

6.1.8.4 FTA_SSL.4 User-initiated Termination

FTA_SSL.4.1 The TSF shall allow Administrator-initiated termination of the Administrator's own interactive session.

Assurance Activity: The evaluator shall perform the following test:

- Test 1: The evaluator initiates an interactive local session with the TOE. The evaluator then follows the operational guidance to exit or log off the session and observes that the session has been terminated.
- Test 2: The evaluator initiates an interactive remote session with the TOE. The evaluator then follows the operational guidance to exit or log off the session and observes that the session has been terminated.

6.1.8.5 FTA_TAB.1 Default TOE Access Banners

FTA_TAB.1.1 Before establishing an administrative user session the TSF shall display a Security Administrator-specified advisory notice and consent warning message regarding use of the TOE.

PP Application Note: *This requirement is intended to apply to interactive sessions between a human user and a TOE. IT entities establishing connections or programmatic connections (e.g., remote procedure calls over a network) are not required to be covered by this requirement.*

Assurance Activity: The evaluator shall check the TSS to ensure that it details each method of access (local and remote) available to the administrator (e.g., serial port, SSH, HTTPS). The evaluator shall also perform the following test:

- Test 1: The evaluator follows the operational guidance to configure a notice and consent warning message. The evaluator shall then, for each method of access specified in the TSS, establish a session with the TOE. The evaluator shall verify that the notice and consent warning message is displayed in each instance.

6.1.8.6 FTA_TSE.1 TOE Session Establishment

FTA_TSE.1.1 The TSF shall be able to deny establishment of a remote VPN client session based on location, time, day.

EP Application Note: *For the EP, location is defined as the clients IP address.*

6.1.8.7 FTA_VCM_EXT.1 VPN Client Management

FTA_VCM_EXT.1.1 The TSF shall assign a private IP address to a VPN client upon successful establishment of a security session.

EP Application Note: *For this requirement, the private IP address is one that is internal to the trusted network for which the TOE is the headend.*

6.1.9 Trusted Path/Channels (FTP)

6.1.9.1 FTP_ITC.1 Inter-TSF-trusted channel

FTP_ITC.1.1 The TSF shall use IPsec, TLS to provide a trusted communication channel between itself and authorized IT entities supporting the following capabilities: audit server, **no other capabilities** 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.[5]

FTP_ITC.1.2 The TSF shall permit the TSF, or the authorized IT entities to initiate communication via the trusted channel.

FTP_ ITC.1.3 The TSF shall initiate communication via the trusted channel for **Syslog**.

EP Application Note: *The NDPP allows trusted channels other than IPsec to be available for communication with external IT entities. To be compliant with the EP, the selection is made such that the TOE must provide the IPsec protocol as a configurable option to the administrator.*

PP Application Note: *The intent of the above requirement is to use a cryptographic protocol to protect external communications with authorized IT entities that the TOE interacts with to perform its functions. This is not, however, to be used to specify VPN gateway functionality; a separate VPN Protection Profile should be used in these instances. Protection (by one of the listed protocols) is required at least for communications with the server that collects the audit information. If it communicates with an authentication server (e.g., RADIUS), then the ST author chooses "authentication server" in FTP_ITC.1.1 and this connection must be protected by one of the listed protocols. If other authorized IT entities (e.g., NTP server) are protected, the ST author makes the appropriate assignments (for those entities) and selections (for the protocols that are used to protect those connections). After the ST author has made the selections, they are to select the detailed requirements in Annex C corresponding to their protocol selection to put in the ST. To summarize, the connection to an external audit collection server is required to be protected by one of the listed protocols. If an external authentication server is supported, then it is required to protect that connection with one of the listed protocols. For any other external server, external communications are not required to be protected, but if protection is claimed, then it must be protected with one of the identified protocols.*

While there are no requirements on the party initiating the communication, the ST author lists in the assignment for FTP_ITC.1.3 the services for which the TOE can initiate the communication with the authorized IT entity.

The requirement implies that not only are communications protected when they are initially established, but also on resumption after an outage. It may

[5] This has been updated to be consistent with TD0035.

be the case that some part of the TOE setup involves manually setting up tunnels to protect other communication, and if after an outage the TOE attempts to re-establish the communication automatically with (the necessary) manual intervention, there may be a window created where an attacker might be able to gain critical information or compromise a connection.

Assurance Activity: The evaluator shall examine the TSS to determine that, for all communications with authorized IT entities identified in the requirement, each communications mechanism is identified in terms of the allowed protocols for that IT entity. The evaluator shall also confirm that all protocols listed in the TSS are specified and included in the requirements in the ST. The evaluator shall confirm that the operational guidance contains instructions for establishing the allowed protocols with each authorized IT entity, and that it contains recovery instructions should a connection be unintentionally broken. The evaluator shall also perform the following tests:

- Test 1: The evaluators shall ensure that communications using each protocol with each authorized IT entity is tested during the course of the evaluation, setting up the connections as described in the operational guidance and ensuring that communication is successful.
- Test 2: For each protocol that the TOE can initiate as defined in the requirement, the evaluator shall follow the operational guidance to ensure that in fact the communication channel can be initiated from the TOE.
- Test 3: The evaluator shall ensure, for each communication channel with an authorized IT entity, the channel data is not sent in plaintext.
- Test 4: The evaluators shall, for each protocol associated with each authorized IT entity tested during test 1, the connection is physically interrupted. The evaluator shall ensure that when physical connectivity is restored, communications are appropriately protected.

Further assurance activities are associated with the specific protocols.

6.1.9.2 FTP_TRP.1 Trusted Path

FTP_TRP.1.1 The TSF shall use SSH provide a trusted communication path between itself and remote administrators that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from disclosure and detection of modification of the communicated data.

FTP_TRP.1.2 The TSF shall permit remote administrators to initiate communication via the trusted path.

FTP_TRP.1.3 The TSF shall require the use of the trusted path for initial administrator authentication and all remote administration actions.

PP Application Note: *This requirement ensures that authorized remote administrators initiate all communication with the TOE via a trusted path, and that all communications with the TOE by remote administrators is performed over this path. The data passed in this trusted communication channel are encrypted as defined the protocol chosen in the first selection. The ST author chooses the mechanism or mechanisms supported by the TOE, and then ensures the detailed requirements in Annex C corresponding to their selection are copied to the ST if not already present.*

Assurance Activity: The evaluator shall examine the TSS to determine that the methods of remote TOE administration are indicated, along with how those communications are protected. The evaluator shall also confirm that all protocols listed in the TSS in support of TOE administration are consistent with those specified in the requirement, and are included in the requirements in the ST. The evaluator shall confirm that the operational guidance contains instructions for establishing the remote administrative sessions for each supported method. The evaluator shall also perform the following tests:

- Test 1: The evaluators shall ensure that communications using each specified (in the operational guidance) remote administration method is tested during the course of the evaluation, setting up the connections as described in the operational guidance and ensuring that communication is successful.
- Test 2: For each method of remote administration supported, the evaluator shall follow the operational guidance to ensure that there is no available interface that can be used by a remote user to establish a remote administrative sessions without invoking the trusted path.
- Test 3: The evaluator shall ensure, for each method of remote administration, the channel data are not sent in plaintext.

Further assurance activities are associated with the specific protocols.

6.2 Security Assurance Requirements

This Security Target conformant with the assurance requirements specified in the NDPP and EP. The CC Part 3 conformant security assurance requirements are listed in Table 8. The CC Part 3 extended assurance requirements are listed in Section 6.1 as “Assurance Activity” and Section 6.2.1.

Table 8: Assurance Requirements

Assurance Class	Assurance Component	Assurance Components Description
Development	ADV_FSP.1	Basic Functional Specification
Guidance Documents	AGD_OPE.1	Operational user guidance
	AGD_PRE.1	Preparative User guidance
Life-cycle Support	ALC_CMC.1	Labeling of the TOE
	ALC_CMS.1	TOE CM coverage
Tests	ATE_IND.1	Independent testing - conformance
Vulnerability Assessment	AVA_VAN.1	Vulnerability analysis

6.2.1 Extended Security Assurance Requirements

These requirements are taken directly from the NDPP and augment or modify the existing SARs taken from CC Part 3.

6.2.1.1 ADV_FSP.1 Basic Functional Specification

There are no specific assurance activities associated with these SARs. The functional specification documentation is provided to support the evaluation activities described in Section 6.1, and other activities described for AGD, ATE, and AVA SARs. The requirements on the content of the functional specification information is implicitly assessed by virtue of the other assurance activities being performed; if the evaluator is unable to perform an activity because the there is insufficient interface information, then an adequate functional specification has not been provided.

6.2.1.2 AGD_OPE.1 Operational User Guidance

Some of the contents of the operational guidance will be verified by the assurance activities in Section 6.1 and evaluation of the TOE according to the CEM. The following additional information is also required.

The operational guidance shall at a minimum list the processes running (or that could run) on the TOE in its evaluated configuration during its operation that are capable of processing data received on the network interfaces (there are likely more than one of these, and this is not limited to the process that "listens" on the network interface). It is acceptable to list all processes running (or that could run) on the TOE in its evaluated configuration instead of attempting to determine just those that process the network data. For each process listed, the administrative guidance will contain a short (e.g., one- or two-line) description of the process' function, and the privilege with which the service runs. "Privilege" includes the hardware privilege level (e.g., ring 0, ring 1), any software privileges specifically associated with the process, and the privileges associated with the user role the process runs as or under.

The operational guidance shall contain instructions for configuring the cryptographic engine associated with the evaluated configuration of the TOE. It shall provide a warning to the administrator that use of other cryptographic engines was not evaluated nor tested during the CC evaluation of the TOE.

The documentation must describe the process for verifying updates to the TOE, either by checking the hash or by verifying a digital signature. The evaluator shall verify that this process includes the following steps:

1. For hashes, a description of where the hash for a given update can be obtained. For digital signatures, instructions for obtaining the certificate that will be used by the FCS_COP.1(2) mechanism to ensure that a signed update has been received from the certificate owner. This may be supplied with the product initially, or may be obtained by some other means.
2. Instructions for obtaining the update itself. This should include instructions for making the update accessible to the TOE (e.g., placement in a specific directory).
3. Instructions for initiating the update process, as well as discerning whether the process was successful or unsuccessful. This includes generation of the hash/digital signature.

The TOE will likely contain security functionality that does not fall in the scope of evaluation under the NDPP. The operational guidance shall make it clear to an administrator which security functionality is covered by the evaluation activities.

6.2.1.3 AGD_PRE.1 Preparative Procedures

As indicated in the introduction above, there are significant expectations with respect to the documentation-especially when configuring the operational environment to support TOE functional requirements. The evaluator shall check to ensure that the guidance provided for the TOE adequately addresses all platforms claimed for the TOE in the ST.

6.2.1.4 ALC_CMC.1 Labeling of the TOE

The evaluator shall check the ST to ensure that it contains an identifier (such as a product name/version number) that specifically identifies the version that meets the requirements of the ST. The evaluator shall ensure that this identifier is sufficient for an acquisition entity to use in procuring the TOE (including the appropriate administrative guidance) as specified in the ST. Further, the evaluator shall check the AGD guidance and TOE samples received for testing to ensure that the version number is consistent with that in the ST. If the vendor maintains a web site advertising the TOE, the evaluator shall examine the information on the web site to ensure that the information in the ST is sufficient to distinguish the product.

6.2.1.5 ATE_IND.1 Independent Testing - Conformance

The evaluator shall prepare a test plan and report documenting the testing aspects of the system. The test plan covers all of the testing actions contained in the CEM and the body of the NDPP's Assurance Activities. While it is not necessary to have one test case per test listed in an Assurance Activity, the evaluator must document in the test plan that each applicable testing requirement in the ST is covered.

The test plan identifies the platforms to be tested, and for those platforms not included in the test plan but included in the ST, the test plan provides a justification for not testing the platforms. This justification must address the differences between the tested platforms and the untested platforms, and make an argument that the differences do not affect the testing to be performed. It is not sufficient to merely assert that the differences have no affect; rationale must be provided. If all platforms claimed in the ST are tested, then no rationale is necessary.

The test plan describes the composition of each platform to be tested, and any setup that is necessary beyond what is contained in the AGD documentation. It should be noted that the evaluator is expected to follow the AGD documentation for installation and setup of each platform either as part of a test or as a standard pre-test condition. This may include special test drivers or tools. For each driver or tool, an argument (not just an assertion) should be provided that the driver or tool will not adversely affect the performance of the functionality by the TOE and its platform. This also includes the configuration of the cryptographic engine to be used. The cryptographic algorithms implemented by this engine are those specified by the NDPP and used by the cryptographic protocols being evaluated (IPsec, TLS/HTTPS, SSH).

The test plan identifies high-level test objectives as well as the test procedures to be followed to achieve those objectives. These procedures include expected results. The test report (which could just be an annotated version of the test plan) details the activities that took place when the test procedures were executed, and includes the actual results of the tests. This shall be a cumulative account, so if there was a test run that resulted in a failure; a fix installed; and then a successful re-run of the tests, the report would show a "fail" and "pass" result (and the supporting details), and not just the "pass" result.

6.2.1.6 AVA_VAN.1 Vulnerability Assessment

As with ATE_IND, the evaluator shall generate a report to document their findings with respect to this requirement. This report could physically be part of the overall test report mentioned in ATE_IND, or a

separate document. The evaluator performs a search of public information to determine the vulnerabilities that have been found in network infrastructure devices and the implemented communication protocols in general, as well as those that pertain to the particular TOE. The evaluator documents the sources consulted and the vulnerabilities found in the report. For each vulnerability found, the evaluator either provides a rationale with respect to its non-applicability, or the evaluator formulates a test (using the guidelines provided in ATE_IND) to confirm the vulnerability, if suitable. Suitability is determined by assessing the attack vector needed to take advantage of the vulnerability. For example, if the vulnerability can be detected by pressing a key combination on boot-up, a test would be suitable at the assurance level of the NDPP. If exploiting the vulnerability requires expert skills and an electron microscope, for instance, then a test would not be suitable and an appropriate justification would be formulated.

~~The evaluator shall generate network packets that cycle through all of the values for attributes, Type, Code, and Transport Layer Protocol, that are undefined by the RFC for each of the protocols, ICMPv4, ICMPv6, IPv4, and IPv6. For example, ICMPv4 has an eight-byte field for Type and an eight-byte field for the Code. Only 21 Types are defined in the RFC (see table 4-2), but there are 256 possible value. Each Type has a Code associated with it, the number of RFC defined Codes varies based on the Type. The evaluator is required to construct packets that exercise each possible value not defined in the RFC (the defined values are already tested in FPF_RUL_EXT.1.10) of Type and Code (including all possible combinations) and target each distinct interface type to determine that the TOE handles these packets appropriately. Since none of these packets will match a rule, or belong to an allowed session the packets should be dropped. Since there are no requirements that the firewall audit a packet being dropped under these circumstances, the evaluator shall ensure the firewall does not allow these packets to flow through the TOE.~~[6]

The evaluator shall generate network packets that cycle through all of the values for the Transport Layer Protocol attribute that are undefined by the RFCs for IPv4 ~~***and IPv6***~~. For example, IPv4 has an eight-bit field for Transport Layer Protocol. Only 100 Transport Layer Protocol values are defined in the RFC for IPv4 (see Table 9-1 in Appendix E), but there are 256 possible values. The evaluator is required to construct packets that exercise each possible value not defined in the RFC (the defined values are already tested in FPF_RUL_EXT.1.7) of Transport Layer Protocol (including all possible combinations) and target each distinct interface type to determine that the TOE handles these packets appropriately. Since none of these packets will match a rule, or belong to an allowed session the packets should be dropped. Since there are no requirements that the VPN gateway audit a packet being dropped under these circumstances, the evaluator shall ensure the VPN gateway does not allow these packets to flow through the TOE. Note that for IPv6, protocol numbers 0 (Hop-by-Hop options), 60 (Destination options), 44 (Fragment), 51 (AH), and 50 (ESP) are extension header numbers rather than transport layer protocol numbers and should be excluded from testing.[7]

In addition to the undefined attribute testing required above, the evaluator shall perform intelligent fuzz testing of the remaining fields in the required protocol headers (excluding FTP). The intent of intelligent fuzzing is that a packet that is otherwise correctly constructed, such that it will be denied when the ruleset is applied, has random values inserted into each of the protocol header fields. The evaluator

[6] Deleted according to TID0013: AVA_VAN.1 in VPN GW EP.

[7] Added according to TID0013: AVA_VAN.1 in VPN GW EP.

ensures a statistically significant sample size, which will vary depending on the protocol field length, is used and is justified in their report.

The evaluator should consult whatever diagnostics (e.g., logging, process status, interface errors) the TOE offers to determine if the TOE was adversely impacted by the processing of such packets.

6.3 Security Requirements Rationale

6.3.1 Security Function Requirement to Security Objective Rationale

The following sections present the rationale that demonstrate that the SFRs meet all security objectives for the TOE.

6.3.1.1 Protected Communications

O.PROTECTED_COMMUNICATIONS

To address the issues concerning transmitting sensitive data to and from the TOE described in Section 3.1, Table 1, row “T.UNAUTHORIZED_ACCESS”, compliant TOEs will provide encryption for these communication paths between themselves and the endpoint. These channels are implemented using one (or more) of three standard protocols: IPsec, TLS/HTTPS, and SSH. These protocols are specified by RFCs that offer a variety of implementation choices. Requirements have been imposed on some of these choices (particularly those for cryptographic primitives) to provide interoperability and resistance to cryptographic attack. While compliant TOEs must support all of the choices specified in the ST, they may support additional algorithms and protocols. If such additional mechanisms are not evaluated, guidance must be given to the administrator to make clear the fact that they are not evaluated.

In addition to providing protection from disclosure (and detection of modification) for the communications, each of the protocols described in this document (IPsec, SSH, and TLS/HTTPS) offer two-way authentication of each endpoint in a cryptographically secure manner, meaning that even if there was a malicious attacker between the two endpoints, any attempt to represent themselves to either endpoint of the communications path as the other communicating party would be detected. The requirements on each protocol, in addition to the structure of the protocols themselves, provide protection against replay attacks such as those described in Section 3.1, Table 1, row “T.UNAUTHORIZED_ACCESS”, usually by including a unique value in each communication so that replay of that communication can be detected.

(FCS_CKM.1, FCS_CKM_EXT.4, FCS_COP.1(1), FCS_COP.1(2), FCS_COP.1(3), FCS_COP.1(4), FCS_IPSEC_EXT.1, FCS_TLS_EXT.1, FCS_SSH_EXT.1, FCS_RBG_EXT.1, FPT_SKP_EXT.1, FTP_ITC.1, FTP_TRP.1)

6.3.1.2 Verifiable Updates

O.VERIFIABLE_UPDATES

As outlined in Section 3.1, Table 1, row “T.UNAUTHORIZED_UPDATE”, failure by the Security Administrator to verify that updates to the system can be trusted may lead to compromise of the entire system. A first step in establishing trust in the update is to publish a hash of the update that can be verified by the System Administrator prior to installing the update. In this way, the Security Administrator can download the update, compute the hash, and compare it to the published hash. While this establishes that the update downloaded is the one associated with the published hash, it does not indicate if the source of the update/hash combination has been compromised or cannot be trusted. So, there remains a threat to the system. To establish trust in the source of the updates, the

system can provide cryptographic mechanisms and procedures to procure the update, check the update cryptographically through the TOE-provided digital signature mechanism, and install the update on the system. While there is no requirement that this process be completely automated, administrative guidance documentation will detail any procedures that must be performed manually, as well as the manner in which the administrator ensures that the signature on the update is valid.

(FPT_TUD_EXT.1, FCS_COP.1(2), FCS_COP.1(3))

6.3.1.3 System Monitoring

O.SYSTEM_MONITORING

In order to assure that information exists that allows Security Administrators to discover intentional and unintentional issues with the configuration and/or operation of the system as discussed in Section 3.1; Table 1; rows “T.ADMIN_ERROR”, “T_UNDETECTED_ACTIONS”, and “T.UNAUTHROIZED_ACCESS”; compliant TOEs have the capability of generating audit data targeted at detecting such activity. Auditing of administrative activities provides information that may hasten corrective action should the system be configured incorrectly. Audit of select system events can provide an indication of failure of critical portions of the TOE (e.g., a cryptographic provider process not running) or anomalous activity (e.g., establishment of an administrative session at a suspicious time, repeated failures to establish sessions or authenticate to the system) of a suspicious nature.

In some instances, there may be a large amount of audit information produced that could overwhelm the TOE or administrators in charge of reviewing the audit information. The TOE must be capable of sending audit information to an external trusted entity, which mitigates the possibility that the generated audit data will cause some kind of denial of service situation on the TOE. This information must carry reliable timestamps, which will help order the information when sent to the external device.

Loss of communication with the audit server is problematic. While there are several potential mitigations to this threat, the NDPP does not mandate that a specific action takes place; the degree to which this action preserves the audit information and still allows the TOE to meet its functionality responsibilities should drive decisions on the suitability of the TOE in a particular environment.

(FAU_GEN.1, FAU_GEN.2, FAU_STG_EXT.1, FPT_STM.1)

O.SYSTEM_MONITORING

EP Application Note: To address the issues of administrators being able to monitor the operations of the VPN gateway, this security objective, which originated in the NDPP, is extended as follows.

Compliant TOEs will implement the ability to log the flow of network traffic. Specifically, the TOE will provide the means for administrators to configure packet filtering rules to ‘log’ when network traffic is found to match the configured rule. As a result, matching a rule configured to ‘log’ will result in informative event logs whenever a match occurs. In addition, the establishment of security associations (SAs) is auditable, not only between peer VPN gateways, but also with certification authorities (CAs).

(FAU_GEN.1, FPF_RUL_EXT.1)

6.3.1.4 TOE Administration

O.TOE_ADMINISTRATION, O.SESSION_LOCK

In order to provide a trusted means for administrators to interact with the TOE, the TOE provides a password-based logon mechanism. The administrator must have the capability to compose a strong password, and have mechanisms in place so that the password must be changed regularly. To avoid

attacks where an attacker might observe a password being typed by an administrator, passwords must be obscured during logon. Session locking or termination must also be implemented to mitigate the risk of an account being used illegitimately. Passwords must be stored in an obscured form, and there must be no interface provided for specifically reading the password or password file such that the passwords are displayed in plain text.

(FIA_UIA_EXT.1, FIA_PMG_EXT.1, FIA_UAU.7, FMT_MTD.1, FMT_SMF.1, FMT_SMR.2, FPT_APW_EXT.1, FTA_SSL_EXT.1, FTA_SSL.3)

O.TOE_ADMINISTRATION

EP Application Note: To address the issues involved with a trusted means of administration of the VPN gateway, this security objective, which originated in the NDPP, is extended as follows. Note that it is assumed that use of the functions indicated below is protected in accordance with the requirements in the NDPP.

Compliant TOEs will provide the functions necessary for an administrator to configure the packet filtering rules, as well as the cryptographic aspects of the IPsec protocol that are enforced by the TOE.

(FMT_SMF.1, FIA_AFL.1)

O.DISPLAY_BANNER

In order to satisfy the policy requiring users to view and consent to an initial access banner prior to accessing the TOE, the TSF displays an Administrator specified advisory notice and consent warning message prior to the establishment of an administrative user session.

FTA_TAB.1

6.3.1.5 Residual Information Clearing

O.RESIDUAL_INFORMATION_CLEARING

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

(FDP_RIP.2)

6.3.1.6 TSF Self-Test

O.TSF_SELF_TEST

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

(FPT_TST_EXT.1)

6.3.1.7 Data Encryption and Decryption

O.CRYPTOGRAPHIC_FUNCTIONS

To address the issues associated with unauthorized disclosure of information, inappropriate access to services, misuse of services, disruption of services, and network-based reconnaissance, compliant TOE’s

will implement a cryptographic capabilities. These capabilities are intended to maintain confidentiality and allow for detection and modification of data that is transmitted outside of the TOE.

(FCS_COP.1(1), FCS_COP.1(2), FCS_COP.1(3), FCS_COP.1(4), FCS_RBG_EXT.1, FCS_IPSEC_EXT.1)

6.3.1.8 Authentication

O. AUTHENTICATION

To further address the issues associated with unauthorized disclosure of information, a compliant TOE's authentication ability (IPSec) will allow a VPN peer to establish VPN connectivity with another VPN peer. VPN endpoints authenticate each other to ensure they are communicating with an authorized external IT entity.

(FTP_ITC.1, FCS_IPSEC_EXT.1)

6.3.1.9 Address-Based Filtering

O.ADDRESS_FILTERING

To address the issues associated with unauthorized disclosure of information, inappropriate access to services, misuse of services, disruption or denial of services, and network-based reconnaissance, compliant TOE's will implement Packet Filtering capability. That capability will restrict the flow of network traffic between protected networks and other attached networks based on network addresses of the network nodes originating (source) and/or receiving (destination) applicable network traffic as well as on established connection information.

(FPF_RUL_EXT.1)

6.3.1.10 Insecure Operations

O. FAIL_SECURE

There may be instances where the TOE's hardware malfunctions or the integrity of the TOE's software is compromised, the latter being due to malicious or non-malicious intent. To address the concern of the TOE operating outside of its hardware or software specification, the TOE will shut down upon discovery of a problem reported via the self-test mechanism.

(FPT_FLS.1)

6.3.1.11 Port Based Filtering

O. PORT_FILTERING

To further address the issues associated with unauthorized disclosure of information, etc., a compliant TOE's port filtering capability will restrict the flow of network traffic between protected networks and other attached networks based on the originating (source) and/or receiving (destination) port (or service) identified in the network traffic as well as on established connection information.

(FPF_RUL_EXT.1)

6.3.1.12 Client Establishment Constraints

O. CLIENT_ESTABLISHMENT_CONSTRAINTS

To address the concern that a remote client may be compromised and attempt to establish connections with the headend VPN gateway outside of "normal" operations, this objective specifies conditions under

which a remote client may establish connections. The administrator may configure the headend VPN gateway to accept a client's request for a connection based on attributes the administrator feels are appropriate.

(FTA_TSE.1)

6.3.1.13 Remote Session Termination

O. REMOTE_SESSION_TERMINATION

A remote client's session can become vulnerability when there is a lack of activity. This is primarily due to a user walking away from a device that has a remote connection established. While some devices have a "lock screen" or logout capability, they cannot always assumed to be configured or available. To address this concern, a session termination capability is necessary during an administrator specified time period.

(FTA_SSL.3(2))

6.3.1.14 Assigned Private Address

O. ASSIGNED_PRIVATE_ADDRESS

There are instances where a remote client desires secure communication with a gateway that is trusted. While a user may be connected via an untrusted network, it should still be possible to ensure that it can communicate with a known entity that controls the routing of the client's network packets. This can be accomplished by the VPN headend assigning an IP address that the gateway controls, as well as providing a routing point for the client's network traffic.

(FTA_VCM_EXT.1)

6.3.2 Security Functional Requirement Dependency Rationale

This Security Target satisfies the SFR dependency rationale by claiming exact compliance to the validated NDPP and VPNEP.

6.3.3 Security Assurance Requirements Rationale

This Security Target satisfies the SAR dependency rationale by claiming exact compliance to the validated NDPP and VPNEP.

7 TOE Summary Specification

This section provides evaluators and potential consumers of the TOE with a high-level description of each SFR, thereby enabling them to gain a general understanding of how the TOE is implemented. These descriptions are intentionally not overly detailed, thereby disclosing no proprietary information. These sections refer to SFRs defined in Section 6, Security Requirements.

The TOE consists of the following Security Functions:

- Security Audit
- Cryptographic Operations
- User Data Protection
- Security Management
- Extended Requirements
- Packet Filtering
- Protection of the TSF
- TOE Access
- Trusted Path/Channels

7.1 Security Audit

7.1.1 Audit Generation

The TSF generates and formats audit logs according to RFC 5424 and include the Pri, Version, Timestamp, Hostname, App-Name, and Msg fields. The Structured-Data, ProcID, and MsgID fields contain NILVALUE. The Pri, Version, and Hostname fields are not relevant to Common Criteria, but may be used to filter audit records once they have been transmitted from the TSF. The Timestamp field specifies the date/time the audit log was generated down to the nearest second. The App-Name field contains the name of the process that generated the audit log. When there is not an external user associated with the audit event, the App-Name field is the subject identity. The TSF uses the Msg field to fulfill the remaining audit requirements. For user generated audit events, The Msg field includes the user's username or X.509 Distinguished Name. The Msg field includes text that describes the audit event (type of event and success or failure) and includes any additional details listed in Table 7.

The TSF generates the audit records for startup and shutdown of the audit function, the administrative actions described in Section 7.5, and the events listed in Table 7.

For FCS_IPSEC_EXT.1, FCS_TLS_EXT.1, and FCS_SSH_EXT.1, the TSF generates audit records for the following protocol failures:

- IPsec
  - Failure to negotiate algorithms during a handshake
  - Session timeout
  - Session dropped (remote client stops responding)
  - Invalid HMAC or GCM tag received
- SSH
  - Failure to negotiate algorithms during a handshake
  - Session timeout
  - Session dropped (remote client stops responding)
  - Invalid HMAC received

- TLS
  - Failure to negotiate algorithms during a handshake
  - Session dropped (remote server stops responding)
  - Invalid HMAC received

The TSF implements specific logging features for the following SFRs:

- FIA_UIA_EXT.1: The TSF logs the origin of local console authentication attempts as “console” For authentication attempts performed over SSH, the TSF logs the origin as the remote IP address of the attempt.
- FPF_RUL_EXT.1: The TSF generates the audit records for each packet filter firewall LOG rule that is configured. These audit records include the network interface, source IP address, destination IP address, transport layer protocol, source port, destination port, and the action taken (i.e. ACCEPT, DROP, or LOG). If a network interface of the TSF receives network traffic faster than it can process it, it drops traffic when its receive queue grows beyond 256 outstanding packets. The TSF keeps track of how many packets have been dropped and logs the number packets that have been dropped over the past minute.

FAU_GEN.1, FAU_GEN.2

7.1.2 Audit Storage

The TSF functions as an Originator and transmits audit logs to a Collector (syslog server) using Syslog over TCP as specified in RFCs 5424, 5425, and 6587. RFC 5424 specifies how the TSF formats logs for local storage and for transmission to the syslog server. The TSF sends audit records to the syslog server simultaneously with the local logging operation. The TSF uses TLS, as specified in Section 7.2.5, to secure the connection with the syslog server. If the link to the syslog server is down and cannot be established, the TSF will continue to store audit records locally. When the syslog server becomes operational, the TSF resumes transmitting new audit logs to the server. The logs generated while the syslog server was unavailable are not transmitted to the syslog server.

The TSF stores local logs in /var/log. The Pri and Version fields are not recorded in the local log files. The TSF stores the audit logs described in Section 7.1.1 as the following discrete local log types: iptables, secure, messages, cprd, quicksec, syslog, zebra, diag, boot, healthd, sysman, ospf, ntpd, common. The TSF command line interface does not provide functions for users to directly access the /var/log directory directly to prevent unauthorized access, modification, or deletion of the logs. The command line interface allows authorized users to view logs via a “show log” command, which provides the user with a read only interface to the log files.

/var/log is a dedicated 10GB partition for local audit log storage. If free space on this partition is exhausted, the TSF will consider this a component failure as described in Section 7.6.1. For each log type, the logging subsystem of the TSF performs log rotation based on time or file size, whichever comes first. Time based rotation occurs daily at an administrator-configured time. File size rotation occurs when the log files reach an administrator-configured size from 1MB to 1000MB, default 100MB. Each local log type is rotated independently of the other local log types. When rotating logs, the TSF compresses the active log file, creates a new log file, and checks to see if the maximum number of archives has been exceeded. If the maximum number of archives has been exceeded, the TSF deletes the oldest archive. The TSF keeps a default of seven archives.

FAU_STG_EXT.1

7.2 Cryptographic Operations

The TSF contains the Red Hat Enterprise Linux Kernel Crypto Module, the OpenSSL v2.0.5 FIPS Crypto Module, and the QuickSec Crypto Library. The crypto modules are certified as follows:

- Red Hat Enterprise Linux Kernel Crypto Module (not FIPS 140-2 certified)
  - AES-GCM (Cert #2983)
- OpenSSL FIPS Object Module v2.0.5 (FIPS 140-2 Cert #1747)[8]
  - AES (Cert #2484[9])
  - SHA-1, 224, 256, 384, 512 (Cert #2102)
  - HMAC SHA-1, 224, 256, 384, 512 (Cert #1526)
  - CTR_DRBG (AES-256) (Cert #342)
  - RSA (Cert #1273)
  - ECDSA (Cert #413)
- QuickSec (not FIPS 140-2 certified)
  - CTR_DRBG (AES-256) (Cert #570)

FCS_COP.1(1), FCS_COP.1(2), FCS_COP.1(3), FCS_COP.1(4), FCS_RBG_EXT.1

7.2.1 Cryptographic Key Generation

The TSF generates ephemeral Diffie-Hellman (FFC) keys for SSH and IKEv2 key exchange according to the following sections of NIST SP 800-56A [9]:

- 5.5.3: Domain Parameter Management
- 5.6.1.1: FFC Key Pair Generation
- 5.6.2.1: Owner Assurances of Static Public Key Validity
- 6.1.2.1: dhEphem, C(2, 0, FFC DH)

The TSF generates Elliptic Curve Diffie-Hellman (ECDH) keys for IKEv2 key exchange according to the following sections of NIST SP 800-56A [9]:

- 5.5.2: Assurances of Domain Parameter Validity
  - Domain parameters are validated according to option 3
- 5.5.3: Domain Parameter Management
- 5.6.1.2: ECC Key Pair Generation
- 6.1.2.2: Ephemeral Unified Model, C(2, 0, ECC CDH)

The TSF implements all “shall” and “should” statements in the referenced sections and does not implement any “shall not” or “should not” in the referenced sections. The TSF does not implement any TOE specific extensions.

FCS_CKM.1(1)

---

[8] The OpenSSL FIPS Object Module v2.0.5 has been ported from RHEL 6 running on Intel Xeon E3-1220v2 (64-bit under vSphere) to Red Hat Enterprise Linux 6.4 running on Intel® Xeon® processor E5-2600 or E5-2600v2 per FIPS 140-2 Implementation Guidance G.5.

[9] The algorithm certificate does not contain any environments that match the OS and Processors used by the TSF; however, the algorithm certificate is considered valid due to the porting of the entire FIPS 140-2 validated module.

The TSF generates 2048-bit RSA keys used for IKEv2 peer authentication and TLS Client Certificate authentication according to the following sections of ANSI X9.31-1998:

- B.4: Generation of Primes
- B.2: Miller-Rabin Probabilistic Primality Test
- B.3: Lucas Probabilistic Primality Test

The TSF implements all “shall” and “should” statements in the referenced sections, with the exception of using a PRNG specified by an ANSI X9 standard. The TSF utilizes the SP 800-90 CTR_DRBG (AES) provided by the OpenSSL Cryptographic module instead. The TSF does not implement any “shall not” or “should not” in the referenced sections. The TSF does not implement any TOE specific extensions.

The TSF generates P-256 and P-384 ECDSA keys used for IKEv2 peer authentication according to Appendix B.4.2: Key Pair Generation by Testing Candidates of FIPS PUB 186-4.

These RSA and ECDSA keys are used in Certificate Signing Requests (CSRs) that also use a SHA-1 or SHA-256 hash.

The TOE also allows externally generated key and cert pairs to be installed. They must be combined in a PCKS#12 format to be installed, or can be installed separately in PEM format.

Two key/cert pairs may be generated:

- Server credentials for VPN
- Credentials for SYSLOG-NG client (RSA only)

FCS_CKM.1(2)

7.2.2 Zeroization

The Master Key is the only persistently stored plaintext key in the TSF. The Master Key is zeroized by overwriting the Master Key file with pseudo random data from the DRBG followed by an overwrite with zeros.

With the exception of the Master Key, the TSF stores all plaintext secret and private keys in RAM. The TSF uses a *tmpfs* to use a portion of RAM as a file system. The TSF derives KEKs from the Master Key (see Section 7.7) to decrypt the encrypted persistent keys into the *tmpfs*. The *tmpfs* is also used by the TSF to store the SAs for the active IPsec tunnels. When a tunnel is closed or an SA is re-keyed, the TSF overwrites the file in the *tmpfs* with zeroes without truncating.

All other secret keys, intermediate cryptographic values, and DRBG states are maintained in RAM. These CSPs are:

- Administrator Passwords (when entered)
- IPsec Session Keys
- VPN Gateway Private Key
- SSH Session Keys
- SSH Private RSA Key
- TLS Session Keys
- Syslog X.509 Client private key
- DRBG States
- Master Key
- KEKs

The TSF overwrites these values with zeroes when no longer needed (e.g., key agreement is complete, session has been re-keyed, session has been terminated).

FCS_CKM_EXT.4

7.2.3 Random Bit Generation

The TSF implements a SP 800-90A CTR_DRBG (AES 256) for generating VPN related key material, CAVP DRBG Cert #570. The TSF initially seeds the DRBG with 3072 bits from /dev/random. The TSF re-seeds the DRBG after generating 8192 bits. The re-seed bits are extracted from /dev/urandom for three consecutive reseed operations and then from /dev/random every fourth re-seed operation.

The TSF implements a separate SP 800-90A CTR_DRBG (AES 256) for generating SSH, TLS, and CSR related key material, CAVP DRBG Cert #264. The TSF initially seeds the DRBG with 384 bits from /dev/random.

For both DRBGs, the TSF accumulates entropy for /dev/random and /dev/urandom from the ID Quantique Quantis PCIe hardware entropy generator and software analysis of system interrupts. The TSF ensures that 3072 bits of entropy are available in the Linux entropy pool at all times. If 3072 bits are not available, the TSF re-fills the pool with entropy from the hardware entropy generator. Both /dev/random and /dev/urandom pull data from the entropy pool, the major difference being /dev/urandom will generate pseudo random bits if insufficient entropy exists. The TSF ensures that each DRBG is seeded with data from /dev/random, and that pseudo random data is only used for a limited number of re-seed operations.

FCS_RBG_EXT.1

7.2.4 IPsec

The TSF implements IPsec as specified in RFCs 4301, 4303, 4106, and 3602. The TSF supports IPsec connections operating in tunnel mode.

The TSF uses the Linux iptables service to perform rule-based packet processing instead of the IPsec Security Policy Database (SPD). The Apriva MESA VPN server enforces a “deny” policy for all packet flows (input, output and forward paths). In order to allow data packets to enter or leave the Apriva MESA VPN server, match rules in the inbound and outbound direction must be created. Packets dropped by the final rule may also be logged, if configured. Logging of dropped packets is desirable in situations where a firewall or router is expected to filter packets ahead of the VPN. A dropped packet on the VPN gives an indication that the firewall configuration may be incorrect.

The Linux iptables rules are defined as access policies. Each access policy allows the Apriva MESA VPN server to manage network traffic as it arrives at and leaves the server. Access policies are used to allow or deny the flow of individual data packets by matching packet header contents against an ordered set of ‘match rules’. A match rule defines an action to take for a given packet’s header content, such as to accept, log or deny the packet based on IP addresses, protocol, or port values.

Access policies are managed within the ‘access-policy’ CLI scope, and consist of two types:

- *Rule sets* are ordered collections of match rules that perform basic actions (permit, deny, reject, return, log, and monitor) based on packet header contents. These rules can be written to support the equivalent of traditional SPD actions (e.g. permit → BYPASS, deny → DISCARD, permit UDP 500/4500 → PROTECT).

- *Access lists* are ordered collections of rule sets, as well as other access lists and individual match rules, that are used to set the access policy for a network interface. The administrator can assign an access list to each network interface for each of the following directions:
    - in: The destination IP address matches the associated network interface
    - out: The source IP address matches the associated network interface
    - forward: Neither source nor destination IP address match the associated network interface.

The Apriva MESA VPN server additionally enforces a restriction on the input/output paths for the external (device-facing) interface. Only IPsec (protocols 50/51, UDP ports 500/4500) and ICMP are allowed on that interface, and no forwarded traffic is allowed (i.e. BYPASS). This is non-configurable from the perspective of allowing additional traffic, but an access policy must be created to allow this traffic on the external network interface for device VPN connections to be established.

The TOE applies these access rules first to an incoming IPsec packet, then again to the packet embedded within the IPsec SA when it attempts to exit the TOE. For example, an IPsec packet containing an HTTP request for a device on the internal network would match both of the following access policies in order: 1) allow incoming UDP 500/4500 traffic on the external network, and 2) allow outgoing HTTP traffic to the internal network. Hence, it is the administrator's responsibility to write access policies that apply to both unestablished and established SAs.

The TSF can be configured to use the AES-GCM-128, AES-GCM-256, AES-CBC-128, and AES-CBC-256 algorithms from the Red Hat Kernel Cryptographic Module for encryption and message authentication for IPsec ESP. When AES-CBC is negotiated as the symmetric cipher, the TOE supports HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512.

The TSF only implements IKEv2 as specified in RFCs 6379. The TSF uses AES-CBC-128 or AES-CBC-256 to encrypt the IKEv2 payloads.

The TSF can be configured to use the following SHA-based HMAC algorithms in IKEv2:

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

The TSF supports IKEv2 SA, also known as phase 1, lifetime configuration with a default value of 24 hours. By default, if no packets are processed (excluding IKEv2 packets) by the IKEv2 SA tunnel within the configured IKEv2 SA lifetime, the SA is closed. In this mode, the IKEv2 SA lifetime like acts as an idle timeout value. If any packets were processed through the IKEv2 SA tunnel, the tunnel is re-keyed instead. The administrator can also configure the IKEv2 tunnel to always rekey rather than drop the tunnel when no packets are processed.

The TSF supports DH groups 14, 19, 20 and 24 for use in IKEv2. One or more of these groups may be selected. The TSF will negotiate the algorithms in the following order if multiple are selected: 20, 19, 24, 14. The TSF uses the CTR_DRBG (Cert #570) to generate a 384-bit ephemeral private key (x) used in Diffie-Hellman.

The TSF generates nonces with the CTR_DRBG (Cert #570) that are 256 bits long. The nonces are used in the IKEv2 key exchange for all cipher suites.

The TSF supports RSA and ECDSA x.509 certificates to perform IKEv2 peer authentication. The RSA keys must be 2048 bits or greater and the ECDSA certificates must use “NIST curves” P-256 or P-384.

The following table lists the equivalent symmetric key sizes for the available IKEv2:

Selected Algorithm	Equivalent symmetric key size
Group 14 (2048-bit MODP)	112 bits
Group 24 (2048-bit MODP, 256-bit Prime)	112 bits
Group 19 (256-bit ECP)	128 bits
Group 20 (384-bit ECP)	192 bits

The TSF negotiates the allowed groups with the client in the IKEv2 exchange. The TSF will not allow the client to use any group not selected in the configuration. For example, if the client has selected group 5, the TSF will refuse to connect because the symmetric strength would be less than 112 bits.

The security management interface, described in Section 7.5, ensures that the Key Size(s) configurable for a CHILD_SA are less than or equal to the Key Size(s) configured for the IKEv2 SA. If a client attempts to negotiate a CHILD_SA with a key size that has not been configured on the TSF, the connection will fail with a cipher-suite mismatch.

When authenticating VPN users, the TSF can be configured to prevent access based on remote IP address, time of day, and/or day of week.

The TSF supports the capability of assigning a private IP address to VPN clients upon successful establishment of a session.

FCS_IPSEC_EXT.1, FTA_SSL.3(2), FTA_TSE.1, FTA_VCM_EXT.1

7.2.5 TLS

The TSF implements a TLSv1.2 client according to RFC 5246. THE TSF supports the TLS_RSA_WITH_AES_128_CBC_SHA and TLS_RSA_WITH_AES_256_CBC_SHA cipher suites. The TSF verifies the remote TLS server's certificate is signed by a trusted CA. The TSF authenticates itself to the remote TLS server using its own X.509v3 certificate.

FCS_TLS_EXT.1

7.2.6 SSH

The TSF implements SSHv2 according to RFCs 4251, 4252, 4253, and 4254.

The SSH implementation supports both public-key (SSH_RSA) and password-based access mechanisms.

The TSF supports SSH_RSA keys of 2048 bits. The TSF supports AES-CBC-128 and AES-CBC-256 for encryption, HMAC-SHA1 for integrity, and diffie-hellman-group14-sha1 for key exchange. The use of diffie-hellman-group14-sha1 is hard coded.

If the TSF's implementation of SSH receives an "SSH packet" larger than 32768 bytes from the TCP layer of the network stack, the TSF silently drops the packet.

The TSF terminates SSH sessions after a configurable period of inactivity.

FCS_SSH_EXT.1, FTA_SSL.3

7.3 User Data Protection

The VPN service, provided by QuickSec, of the TSF takes care to zeroize sensitive internal data before the memory is freed or reused. This includes keys, MACs, cipher states, HMACs, and random numbers. Each buffer used to process VPN data is also zeroized before it is freed (user space).

When the TSF receives data from the network, it allocates a buffer equal to the size of the received data. The received data completely overwrites any pre-existing data in the buffer effectively zeroizing the residual data upon allocation.

When the TSF prepares to send data on the network, each process allocates a buffer (which may contain residual data), writes a contiguous block of new data to the buffer, passes the buffer to the network stack, and tells the network stack the amount of data that was written to the buffer. This effectively zeroizes the residual data upon allocation. If the block of data is shorter than the buffer, the end of the buffer may contain residual data; however, the residual data is not sent, because the network stack only sends out the amount of data that was written to the buffer.

In the case of VPN data passing through the TSF, both the VPN zeroization and the standard network zeroization of residual data are applied to the packets.

FDP_RIP.2

7.4 Identification and Authentication

The TSF can be administered through two interfaces, the local console and SSH.

When a user connects to the console interface, the TSF prompts the user for a username and password. The TSF does not echo any characters back to the local console while the user is entering their password. If the username/password match an authorized administrator's credentials, the user is granted access to the command line interface described in Section 7.5.

When a user connects to the SSH interface, the TSF checks to see if the user proposed public key authentication. If the client proposed public key authentication, the TSF attempts to authenticate the user using the username and SSH_RSA (RFC 4253). If the SSH_RSA authentication fails or the client did not propose public key authentication, the TSF attempts to authenticate the client using a username/password. If either SSH_RSA authentication or username/password match an authorized administrator's credentials, the user is granted access to the command line interface described in Section 7.5.

The TSF requires passwords to be 15 characters or greater. The TSF supports passwords containing ASCII characters 0x21 thru 0x7E inclusive.

FIA_UIA_EXT.1, FIA_PMG_EXT.1, FIA_UAU_EXT.2, FIA_UAU.7

The TSF maintains a separate failed authentication counter and lock flag for each remote (SSH) administrative user account. When a user attempts to establish an SSH session, the TSF checks if the lock flag has been set once the user has provided their username. If the account has been locked, the TSF does not process the authentication data and terminates SSH connection. Otherwise, the TSF processes the authentication attempt. A failed public key authentication attempt immediately followed by a failed username/password authentication attempt in the same SSH session is counted as a single failure, because most SSH clients automatically attempt public key authentication. A failed public key authentication attempt that is not followed by a username/password attempt is still counted as a failed authentication attempt. Each failed username/password authentication attempt is individually counted, with the exception of the case noted above. For each unsuccessful authentication attempt, the TSF increments the counter, compares the counter to the configured limit, and sets the lock flag if the counter has reached the configured limit. For each successful authentication attempt, the TSF resets the failed authentication counter to zero. Accounts can be unlocked only by a user over the console connection.

FIA_AFL.1

The TSF stores SSH certificates and public keys in /home/*username*/.ssh. The .ssh directory is readable only by the respective user, but the command-line interface prevents any direct access to the files.

The TSF protects the private keys associated with the TSF's X.509 certificates as described in Section 7.7.

The TSF stores its X.509 certificates, X.509 CA certificates, and CRLs in /etc/apriva/certs. The CA certificates are used to verify the Syslog Server's certificate and the VPN client certificates. The command line interface described in Section 7.5 restricts access to the certificate store to authorized administrative users.

The TSF verifies certificates by checking the following:

- Current date between the "Valid from" and "Valid to" dates
- The Certificate is not listed on the CRLs that have been imported into the TSF
- The certificate path is valid:
    - The certificate is signed by a known/trusted CA
    - The certificate includes the certificates of the intermediate CAs where each successive certificate is signed by the next certificate, with the last certificate being signed by a known/trusted CA.

The TSF verifies the validity of a certificate when an administrator loads a certificate into the TSF, when the TSF loads its certificates into memory, when IKEv2 receives a client certificate, and when Syslog/TLS receives a server certificate. If the administrator attempts to load a certificate with a Subject Type=CA, the TSF does not validate the certificate path.

FIA_X509_EXT.1

7.5 Security Management

The TSF does not allow any administrative actions to be performed prior to authentication of the administrative user. Once the administrative user is authenticated, the TSF grants the user access to a restricted command-line shell. This shell restricts the administrative users to commands required for administering the TSF while preventing users from running general-purpose Linux commands.

The TSF enforces these restrictions by restricting the administrators to a restricted command environment. When the TSF grants access to an administrative user using SSH protocol or the console, the user has read only access to non-sensitive data. Authorized administrators must run a separate "enable" command, enter an additional password, and be assigned the authorized administrator (exec) privilege to gain access to privileged mode. The TSF generates an additional audit record when a user attempts (successes and failures) to access privileged mode. This audit record includes the username, the time and date and location (remote IP address or console). Any changes to the system configuration or stored data can be performed only in privileged mode.

The TSF restricts the following functions to authorized administrators who have been assigned the security administrator role:

- Manage trusted CAs
- Load CRL
- Configure Packet filtering rules (as described in Section 7.6)
- Configure IKEv2 SA lifetimes
- Configure IKEv2 algorithms
- Mange IKEv2 Session Establishment restrictions

- Configure IPsec algorithms
- Configure IP address assignment to VPN clients
- Generate CSR (and RSA or ECDSA key pair)
- Load a X.509 Certificate
- Load a private key (associated with an X.509 certificate)
- Load and assign SSH_RSA public key
- Unlock account (console only)
- Manage administrator accounts
- Manage minimum password length
- Configure the remote administrator inactivity timeout
- Manage the failed authentication counter
- Configure Syslog server connectivity
- Configure NTP server connectivity
- Initiate an update to the software
- Set the time
- Backup and restore key material

Administrative user SSH_RSA keys are loaded onto the TSF via a command that itself uses SSH (scp) to copy the public keys to the TSF from the client's machine. Once the keys is loaded and enabled, the client can SSH into the TSF using the corresponding private key.

FMT_MOF.1, FMT_MTD.1, FMT_SMF.1, FMT_SMR.2

7.6 Packet Filtering

The TSF has four physical network interfaces:

- VPN Ingress (public untrusted network)
- VPN Egress (internal trusted network)
- Management
- Traffic Analysis – disabled by default

The TSF has one virtual network interface:

- VPN Interface

While the TSF is powering up, the TSF does not enable any network interfaces prior to completion of the power-up self-tests. This ensures that the TSF is operating properly and that the packet filtering rules have been initialized before the TSF processes any network data.

The TSF implements three different rule chains that can be applied to network traffic on the VPN Ingress, VPN Egress, Management, or VPN Interface. Each chain is applied to a different traffic type; traffic addressed to the TOE (INPUT), traffic sent by the TOE (OUTPUT), and traffic passing through the TOE (FORWARD). The rules are applied in the order they appear. Each rule can be ACCEPT, DROP, or LOG. Traffic can be filtered by interface (based on the name of the interface), IP protocol (TCP, UDP), port range and IP address range. FORWARD rules are applied to VPN traffic. The FORWARD rules specify the VPN virtual interface instead of the hardware interface, so failure of the VPN results in iptables being unable to send data over the VPN interface.

The TSF implements two hard-coded iptables rules that cannot be modified:

- DROP and LOG any packets that are not matched.

- "integrated" mode. This mode is disabled by default. When enabled, it mirrors all plaintext data sent to or from the VPN interface to the Traffic Analysis interface. This rule cannot be modified, only enabled or disabled. This feature provides for CALEA, FISA and packet statistical analysis.

Packets entering the TSF's network stack on both physical and virtual interfaces are filtered by iptables. iptables examines the following fields within the header of each packet: Source Address (IPv4), Destination address (IPv4), Protocol (IPv4), Source Port (TCP or UDP), and destination port (TCP or UDP). The IPsec engine then performs its own filtering and processing as necessary to encrypt and decrypt packets. Finally, the resulting packets are processed via iptables once more.

The TSF initially sets iptables to block traffic on the ingress and egress ports. When the VPN service starts, it initializes and performs various self-tests. Once complete, the TOE loads iptables with the active VPN configuration to allow VPN traffic to commence.

FPF_RUL_EXT.1

7.6.1 Component Failure

If the TSF detects a failure in the VPN service (i.e., TRNG failure, DRBG failure, Network Interface failure, audit storage exhaustion), the TSF performs a shutdown of VPN service to prevent packets from flowing through the TOE. A TRNG failure or DRBG failure is detected through the use of a continuous random number generator test that compares the current block of output with the previous block. If the two blocks are identical, the test fails. Network Interface failure is detected by an independent health monitoring process, which polls the network interface statistics to determine if the interface is reporting any problems. When the VPN service is shutdown, existing tunnel traffic cannot pass and iptables will continue to block traffic through the VPN. Once the VPN service shutdown has completed, iptables is reconfigured to its initial start-up mode. The health monitor of the VPN will periodically check that iptables is running. If the TSF's health monitoring process sees that iptables is not running it will halt the VPN service and generate an audit record. The health monitor does not check correctness of the filter rules during operation.

FPF_RUL_EXT.1

7.6.2 RFC Conformance

The TSF supports packet filtering on fields in the following protocols:

- RFC 791 (IPv4)
- RFC 793 (TCP)
- RFC 768 (UDP)

Apriva has verified that the TSF correctly implements these protocols through third party interoperability testing. The TSF has been tested to be compatible with the IPv4, TCP, and UPD implementations in Google Android, Linux 2.6, MacOS X, and Microsoft Windows.

Apriva has verified that the TSF correctly implements IPsec (IKEv2 and ESP) through third party interoperability testing. The TSF has been verified to be compatible with strongSwan Linux clients, SimpleVPN Client, and Mocana KeyVPN™ Client software on Android phones. The QuickSec Server, which is part of the TSF, carries the following compliance statement:

> *Standards-compliance. QuickSec Server conforms with the relevant official and industry standards, such as ISO X.509, RSA Laboratories PKCS #1, PKCS #10, NIST Digital Signature Standard (FIPS PUB 186), NIST Data Encryption Standard (FIPS PUB 46-1), and the ANSI C standard.*

Interoperability. The QuickSec Server implementation has been tested for interoperability with the related products of other major vendors involved in IPsec development. Regarding the IKEv2 functionality, interoperability tests have been completed with several ICSA participants. AuthenTec QuickSec Server has participated and successfully undergone the VPNC logotesting program and demonstrated its interoperability against all other participating vendors.

FPF_RUL_EXT.1

7.7 Protection of the TSF

The TSF persistently stores the following secret keys, private keys, and other CSPs in non-plaintext form:

- Administrator passwords – Salted and hashed 5000 times with SHA-256
- VPN Gateway Private Key – encrypted with a 128 bit AES KEK
- SSH Private RSA Host Key – encrypted with a 128 bit AES KEK
- Syslog X.509 Client private key – encrypted with a 128 bit AES KEK

The TSF persistently stores the following secret keys, private keys, and other CSPs in plaintext form:

- Master Key – 128 bit AES key, stored at /etc/apriva/cprd

The CLI, described in Section 7.5, does not provide the user with any commands that allow for the reading of plaintext secret keys, private keys, or CSPs. The TSF implements strict access permissions so only the process that needs to access a CSP has read permissions (e.g., Syslog is the only process with read permissions for the Syslog X.509 Certificate).

The TSF maintains a Master Key that is stored in /etc/apriva/keys. The master key is never displayed or backed up. The master key is generated as follows:

1. The system creates a random Initialization Vector (IV) 128-bits in length. The IV is stored in /etc/apriva/keys.
2. The system prompts the user(s) for two passphrases. Once both passphrases have been entered, they are concatenated together.
3. PBKDF2 function using SHA1 as the pseudorandom function is used to create the Master Key from the IV as the salt and passphrase as the password for 100 iterations. The CLI, described in Section 7.5, does not provide the user with any commands that allow for the reading of the Master Key or the /etc/apriva/keys directory.

Each private key or certificate has a Key Encrypting Key (KEK) associated with it. A KEK is a 128-bit number that is derived from the Master Key and a hardcoded “salt” by running the IV and salt through the SHA-256 hash function. Each key type (syslog, gateway) has its own hard-coded salt value.

This method of key management allows the IV and encrypted keys to be backed up without compromising the security of the master key. The master key can be restored by re-loading the IV and entering the pass phrases.

FPT_SKP_EXT.1, FPT_APW_EXT.1

The following TSF security functions utilize the time:

- Audit timestamps
- IPsec SA timeout
- SSH session timeout
- Console session timeout
- Certificate Expiration/Validity Checking

The TSF contains a real-time clock to maintain the time between updates from the NTP server and provide time to other TSF security functions. The real-time clock is considered reliable, because the TSF security functions that utilize the time only utilize an accuracy of one second.

FPT_STM.1

A support point of contact (POC) will be assigned on a sale by sale basis to match customer requirements. Telephone support, email support, and SW updates will be handled through this POC. Each support POC distributes updates by sending customers CDs containing the updated software.

The TSF utilizes the RedHat RPM package management system to validate and install software updates. The TSF is configured to trust updates that are digitally signed by RedHat's private key and Apriva's private key. The TSF disallows the user from performing an update using a solely Red-Hat signed RPM. The update package must be signed by Apriva, but sub-packages can be signed by Red-Hat only. This ensures that users do not load arbitrary Red Hat RPMs on the TSF. Both the RedHat and Apriva private keys are RSA 2048-bit keys. RedHat-supplied RPM files are digitally signed with RedHat's private key. Apriva-created packages are only signed with an Apriva private key. The TSF automatically verifies the signature of any package that is updated. If the signature verification fails, the TSF logs the failure, aborts the update, and deletes the invalid package. If the signature verification succeeds, the TSF installs the update.

The TSF stored public keys used to verify software update in plaintext on the file system. The TSF's CLI prevents the users from modifying these public keys by preventing direct file system access. The only method of modifying a public key is to use the trusted update function to update the package that contains the key.

FPT_TUD_EXT.1

When the TSF starts-up, it runs the following self-tests:

- Hardware POST
- TRNG Health Test
- SW Crypto-Self Test
- Digital Signature Integrity test of the VPN Server

The Hardware POST consists of basic tests of Power Supply, CPU, memory, BIOS disk, I/O interfaces (USB, Network, etc.). If the TSF encounters a hardware POST error other than a single power supply failure, it results in the TSF shutting down.

The IDQ Quantis TRNG performs its own internal status checks continuously. The TSS health check routine will read 512 bytes of data from the IDQ directly. If the read fails, the TSF generates an audit record and fails the test.

The OpenSSL FIPS Object Module performs a FIPS 140-2 integrity test by performing an HMAC-SHA-1 of the OpenSSL binary and a Known Answer Test (KAT) on each algorithm implemented within it. The QuickSec cryptographic module also performs an integrity test and KAT; however, these tests have not been validated by FIPS 140-2. Each KAT consists of calling the algorithm with known inputs and verifying that the output matches the expected (pre-computed) output.

The TSF verifies the integrity of all TSF components using a cryptographic hash of each binary and config file. The integrity check is performed by running a SHA1 hash of each of the files that are part of the install package:

- executable file

- kernel loadable modules
- configuration
- service startup/shutdown scripts

The TSF compares the hashes against a previously generated list of SHA1 hashes. The list is updated when the TSF is updated (new VPN RPM is installed) or when the user updated the VPN configuration. If any of the hash comparisons fail, the TSF shuts down. If all checks succeed, the TSF proceeds with its standard boot sequence.

The TSF also performs continuous tests of the TRNG, by running the power-on TRNG test once per minute. The TSF terminates the VPN functionality if the TRNG test fails.

FPT_TST_EXT.1, FPT_FLS.1

7.8 TOE Access

The administrator can access the TSF via the local console (serial) or remotely via SSH. The TSF displays a configurable advisory and consent message when administrator accesses the CLI through either interface. The administrator can terminate a CLI session (both local console and SSH) by logging out. The TSF terminates local console sessions after a configurable period of inactivity.

FTA_TAB.1, FTA_SSL.4, FTA_SSL_EXT.1

7.9 Trusted Path/Channels

The TSF communicates with the following trusted IT entities:

- VPN Clients – IPsec
- Syslog Server – TLSv1.2

The TSF implements the trusted channel protocols as described in Sections 7.2.4 and 7.2.5.

FTP_ITC.1

SSHv2 is the only method of remote TOE administration. The TSF only listens for SSH connections on the management network. The TSF implements the trusted path protocol as described in Section 7.2.6.

FTP_TRP.1

8 Terms and Definitions

Table 9: Abbreviations and Acronyms	
Abbreviations / Acronyms	Description
AH	Authentication Header
AES	Advanced Encryption Standard
AGD	Apriva Guidance Document
CA	Certificate Authority
CAC	Common Access Card
CAP	Composed Assurance Package
CAVP	Cryptographic Algorithm Validation Program
CBC	Cipher Block Chaining
CC	Common Criteria
CCRA	Arrangement on the Recognition of Common Criteria Certificates in the field of IT Security
CLI	Command Line Interface
COSP	Code of Standard Practice
CRL	Certificate Revocation List
CSP	Critical Security Parameter
CSR	Certificate Signing Request
CTR	Cooperative Threat Reduction
DAC	Discretionary Access Control
DH	Diffie-Hellman
DN	Distinguished Name
DOD	Department of Defense
DRBG	Deterministic Random Bit Generator
DSA	Digital Signature Algorithm
DSA2VS	Digital Signature Algorithm Validation System
EAL	Evaluation Assurance Level
ECDSA	Elliptic Curve Digital Signature Algorithm
ECDSA2VS	Elliptic Curve Digital Signature Algorithm Validation System
EP	Extended Package
ESP	Encapsulating Security Payload
FIPS	Federal Information Processing Standards
GCM	Galois Counter Mode
HMAC	Hash Message Authentication Code
HTTP	Hypertext Transfer Protocol
I&A	Identification and Authentication
ICMP	Internet Control Message Protocol
IDQ	Internet Data Query
IPsec	Internet Protocol Security (suite of protocols)
IKE	Internet Key Exchange
IPsec	Internet Protocol Security
IT	Information Technology
KEK	Key Encrypting Key
NDPP	Network Device Protection Profile
NIAP	National Information Assurance Partnership
NTP	Network Time Protocol

Table 9: Abbreviations and Acronyms	
Abbreviations / Acronyms	Description
OPE	Outboard Processing Environment
OSP	Organizational Security Policy
PBKDF2	Password-Based Key Derivation Function 2
PP	Protection Profile
rDSA	Reliable Data Security Architecture
RFC	DARPA Internet Engineering Task Force Request for Comments
RS	Radio Sector- RS-232 is a standard for serial communication transmission of data
RSA	Ron Rivest, Adi Shamir and Leonard Adleman,
RSA2VS	RSA Validation System
SA	Security Association
SAR	Security Assurance Requirement
SFP	Security Function Policy
SFR	Security Functional Requirement
SFR	Security Function Requirements
SHAVS	Secure Hash Algorithm Validation System
SPD	Security Policy Database
SSH	Secure SHell
ST	Security Target
TLS	Transport Layer Security
TOE	Target of Evaluation
TRBG	True Random Bit Generator
TSF	TOE Security Functionality
TSFI	TSF Interface
TSS	TOE Summary Specification
VPN	Virtual Private Network
VPNEP	Virtual Private Network Extended Package

9 References

Table 10: TOE Guidance Documentation			
Reference	Description	Version	Date
[1]	Apriva MESA VPN NIAP Guidance	V1.0	May 28, 2015

Table 11: Common Criteria v3.1 References			
Reference	Description	Version	Date
[2]	Common Criteria for Information Technology Security Evaluation Part 1: Introduction and general model CCMB-2009-07-001	V3.1 R3	July 2009
[3]	Common Criteria for Information Technology Security Evaluation Part 2: Security functional components CCMB-2009-07-002	V3.1 R3	July 2009
[4]	Common Criteria for Information Technology Security Evaluation Part 3: Security assurance components CCMB-2009-07-003	V3.1 R3	July 2009
[5]	Common Criteria for Information Technology Security Evaluation Evaluation Methodology CCMB-2009-07-004	V3.1 R3	July 2009

Table 12: Supporting Documentation			
Reference	Description	Version	Date
[6]	Protection Profile for Network Devices	1.1	June 8, 2012
[7]	Security Requirements for Network Devices Errata #3		November 3, 2013
[8]	Network Device Protection Profile (NDPP) Extended Package VPN Gateway	1.1	April 12, 2013
[9]	NIST SP 800-56A, Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography (Revised)		March 2007
[10]	FIPS PUB 186-3, Digital Signature Standard (DSS)		June 2009

[CC_PART1] Common Criteria for Information Technology Security Evaluation –Part 1: Introduction and general model, version 3.1, Revision 4, CCMB-2012-009-001. September 2012. Common Criteria Maintenance Board.

[CC_PART2] Common Criteria for Information Technology Security Evaluation –Part 2: Security functional components, version 3.1, Revision 4, CCMB-2012-009-002. September 2012. Common Criteria Maintenance Board.

[CC_PART3] Common Criteria for Information Technology Security Evaluation –Part 3: Security assurance components, version 3.1, Revision 4, CCMB-2012-009-003. September 2012. Common Criteria Maintenance Board.

[CEM] Common Methodology for Information Technology Security Evaluation –Evaluation Methodology, version 3.1, Revision 4, CCMB-2012-009-004. September 2012. Common Criteria Maintenance Board.

[FIPS 140-2] FIPSPUB 140-2 - Federal Information Processing Standards Publication Security Requirements for Cryptographic Modules. May 25, 2001.

[FIPS PUB 180-3] Federal Information Processing Standards Publication Secure Hash Standard (SHS). October 2008.

[FIPS PUB 186-2] Federal Information Processing Standards Publication. January 27, 2000.

[FIPS PUB 186-3] Federal Information Processing Standards Publication Digital Signature Standard (DSS). June 2009.

[FIPS PUB 197] Federal Information Processing Standards Publication Advanced Encryption Standard (AES). November 26, 2001.

[FIPS PUB 198-1] Federal Information Processing Standards Publication the Keyed-Hash Message Authentication Code (HMAC). July 2008.

[NIST SP 800-38A] NIST Special Publication 800-38A - Recommendation for Block 2001 Edition Recommendation for Block Cipher Modes of Operation Methods and Techniques. December 2001.

[NIST SP 800-56A] NIST Special Publication 800-56A - Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography (Revised), March 2007. The National Institute of Standards and Technology.

[NIST SP 800-56B] NIST Special Publication 800-56B - Recommendation for Pair-Wise, Key Establishment Schemes Using Integer Factorization Cryptography. August 2009.

[NIST SP 800-57] NIST Special Publication 800-57 - Recommendation for Key Management

[NIST SP 800-90A] NIST Special Publication 800-90A - Deterministic Random Bit Generator Validation System (DRBGVS). February 14, 2013.

[pp_nd_v1.1] Protection Profile for Network Devices version 1.1. June 8, 2012.

[pp_nd_v1.1-err3] Security Requirements for Network Devices: Errata #3. November 3, 2014.

[pp_nd_vpn_gw_ep_v1.1] Network Device Protection Profile (NDPP) Extended Package VPN Gateway version 1.1. April 12, 2013.

ANSI X9.31-1998, Digital Signatures Using Reversible Public Key Cryptography for the Financial Services Industry (rDSA), ANSI X9.31-1988, September 1998. Appendix B.4 Generation of Primes