CyberFence 3e-636 Series Network Security Devices (NDcPP10) Security Target

Version 0.7
09/19/2017

3e Technologies International

9715 Key West Avenue
5th Floor
Rockville, MD 20850 USA

www.ultra-3eti.com

LIST OF TABLES

1. Security Target Introduction

This section identifies the Security Target (ST) and Target of Evaluation (TOE) identification, ST conventions, ST conformance claims, and the ST organization. The TOE is CyberFence 3e-636 Series Network Security Devices provided by 3e Technologies International. The TOE is being evaluated as a Network Device.

The Security Target contains the following additional sections:

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

Conventions

The following conventions have been applied in this document:

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

1.1 Security Target Reference

ST Title – CyberFence 3e-636 Series Network Security Devices (NDcPP10) Security Target

ST Version – Version 0.7

ST Date – 09/19/2017

1.2 TOE Reference

TOE Identification – 3e Technologies International CyberFence 3e-636 Series Network Security Devices

TOE Developer – 3e Technologies International

TOE Models – 3e-636L3 EtherGuard, 3e-636L2 DarkNode, 3e-636H UltraCrypt, 3e-636A EtherWatch

TOE Firmware Versions: 5.1.300 for 3e-636L3 EtherGuard, 5.1.300 for 3e-636L2, 3e-636H, 3e-636A

Evaluation Sponsor – 3e Technologies International

Keywords – Encryption, VLAN, VPN, IPsec, access control, data packet inspection, traffic filter

1.3 TOE Overview

The Target of Evaluation (TOE) is CyberFence 3e-636 Series Network Security Devices. 3eTI's 636 Series Network Security Devices offer the multiple capabilities necessary for protecting embedded devices and safety-critical industrial control systems (ICS) against internal and external attacks. The core capabilities include: network access control, OSI Layer 2 and Layer 3 packet filtering, industrial control protocols packet inspection and secured application data transportation (via encryption).

This Security Target only addresses the functions that provide for the security of the TOE itself as described in Section 1.4.1.2 TOE logical boundaries below.

1.3.1 Hardware, Firmware, and Software Required by the TOE

The TOE requires the following Operational Environment support which is not included in the TOE's physical boundary.

- **Administrator Workstations:** Trusted administrators access the TOE through the TLS/HTTPS protocol.
- **Audit Servers:** The TOE relies upon the audit server for storage of audit records. The TOE itself stores limited amount of the audit records in its internal persistence storage. Those audit records are accessible and exportable through the Web GUI interface.
- **NTP Servers:** The TOE relies upon an NTP server to provide reliable time**.** If the time is configured locally, the TOE will use its own reliable hardware clock to maintain time as well.
- **LDAP Server:** The TOE relies on the LDAP server for centralized authentication of administrator if the security administrator chooses this configuration. The TOE can also authenticate administrator using local user name and password.

1.4 TOE Description

The TOE is composed of both hardware and firmware. All four models of the 3e-636 series devices share identical hardware. The firmware contains modules, that when activated, can provide additional functionality to the device. The 3e-636L3 runs firmware with naming convention: "signed_636N-**L3**.5.1.300.00.43.bin". The last 2 digits are associated with the build number for a minor bug fix release that affected non security related functionality associated with this devices capability. 3e-636L2 DarkNode, 3e-636H Ultracrypt and 3e-636A EtherWatch run on firmware with naming convention: "signed_636N.5.1.300.00.50.bin". The evaluated version of the product was the 3e-636L2 DarkNode.

The two sets of firmware are treated as equivalent under the context of this CPPNDV10 validation in that they share a common base consisting of a collection of modules that implement all CPPNDV10 SFRs, those are:

- Same vendor's own OpenSSL library and cryptographic hardware driver
- Same HTTPS module
- Same IPsec module
- Same WebGUI for security functional management
- Same audit log module for CPPNDV10 audit logs

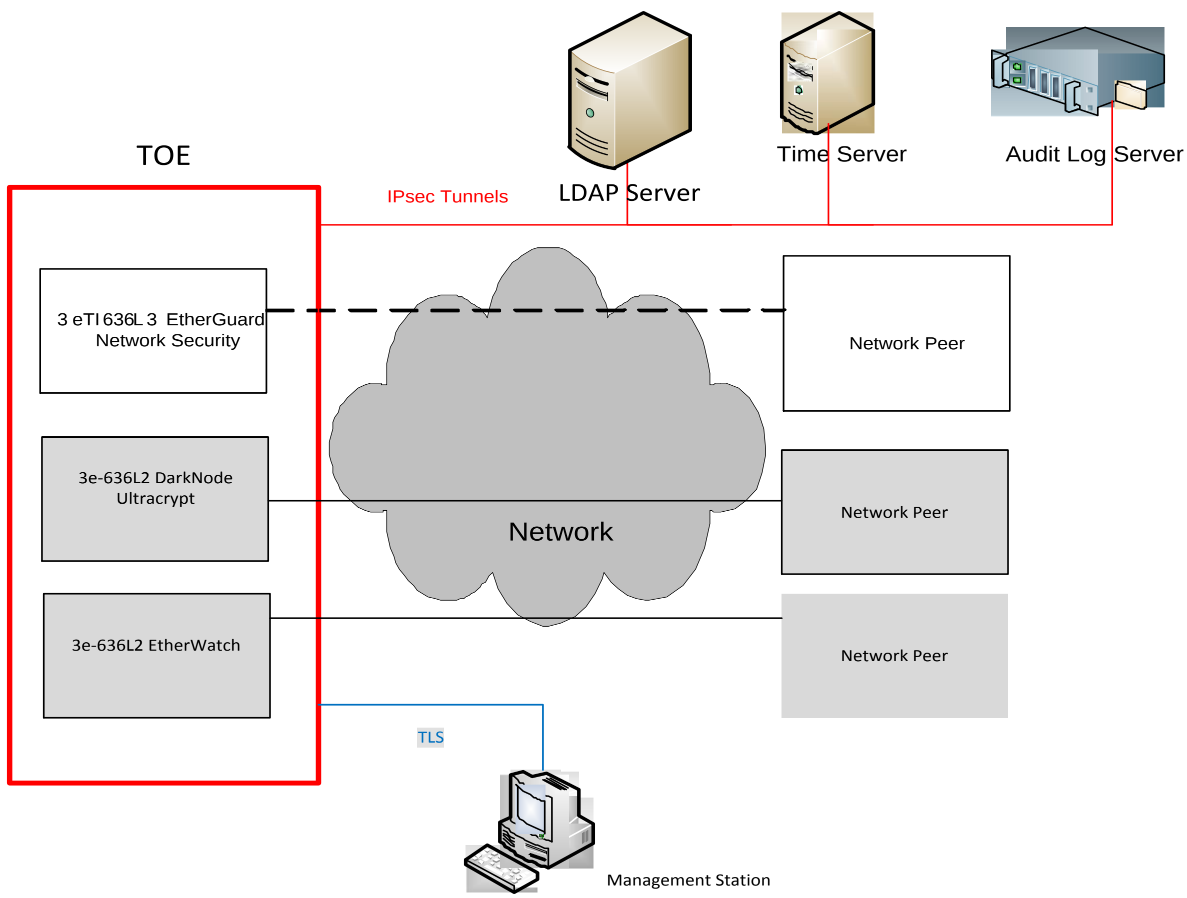
The differences between the two sets of firmware are the addition of other modules which implement functionality that is outside the scope of this evaluation. The 3e-636L3 is the common base plus a network data routing module, firewall modules and Industrial Protocol DPI modules, while the 3e-636L2 is the common base plus a VLAN encryption module, with firewall and industrial protocol DPI as optional modules.

The table below identifies the additional product functional features supported by each model of the TOE. Again, none of the features listed below are included in the evaluated configuration.

Device Name	VPN Encryption	VLAN Encryption	Access Control	Industrial Protocol Packet Inspection
3e-636L3 EtherGuard	X		X	X
3e-636L2 Darknode		X	X	X
3e-636H Ultracrypt		X		
3e-636A EtherWatch			X	X

1.4.1 TOE Architecture

The figure below illustrates the typical deployment use case and operational environment setup for TOE devices.



All devices operate in the same operation environment. IPsec tunnels are used to secure the communication between device and external servers such as NTP server, Audit log server and LDAP server. All devices offer the same HTTPS/TLS based GUI interface for device configuration and management.

1.4.1.1 Physical Boundaries

The TOE physical boundary defines all hardware and firmware that is required to support the TOE's logical boundary and the TOE's security functions. The TOE hardware platform uses FreeScale MPC8378E CPU and the TOE's firmware contains an embedded Linux Kernel customized by 3eTI based on kernel version 4.6. In short, the TOE's physical boundary is the physical device for all models.

The table below describes the ports and interfaces implemented by the TOE

Port/Interfaces	Management/Control I/O	Data Input	Data Output	Status Output	Same on all devices
Local Management Ethernet port (1)	X				Yes
Plain text Ethernet port (1)	X	X	X		Yes
Cipher text Ethernet port (1)	X	X	X		Yes
Auxiliary Ethernet port (1)	N/A	N/A	X		Disabled on 3e-636L3. Bridged to Plain text port on 3e-636L2, 3e-636H and 3e-636A
Power					Yes
LED				X	Yes
Reset Pin	X				Yes

The Operational Environment components relied upon by the TOE and not included in the physical boundary are described in Section 1.3.1.

1.4.1.2 Logical Boundaries

This section summarizes the security functions provided by CyberFence 3e-636:
- Security audit
- Cryptographic support
- Identification and authentication
- Security management
- Protection of the TSF
- TOE access
- Trusted path/channels

1.4.1.2.1 Security audit

The TOE generates auditable events for actions on the TOE with the capability of selective audit record generation. The records of these events can be viewed within the TOE Management Interface or they can be exported to audit systems in the Operational Environment. The TOE generates records for its own actions, containing information about the user/process associated with the event, the success or failure of the event, and the time that the event occurred. Additionally, all administrator actions relating to the management of TSF data and configuration data are logged by the TOE's audit generation functionality.

1.4.1.2.2 Cryptographic support

The TOE uses NIST SP 800-90 DRBG random bits generator and the following cryptographic algorithms: AES, RSA, ECDSA, SHA, HMAC to secure the trusted channel and trusted path communication. The TOE is designed to zeroize Critical Security Parameters (CSPs) to mitigate the possibility of disclosure or modification.

1.4.1.2.3 Identification and authentication

The TOE provides Identification and Authentication security functionality to ensure that all users are properly identified and authenticated before accessing TOE functionality. The TOE enforces a password-based authentication mechanism to perform administrative user authentication. Passwords are obscured when being displayed during any attempted login. Administrative users can be authenticated via either local user database or remote LDAP server. The TOE also authenticates its IPsec peers; the authentication is performed over IKEv2 SA phase of mutual authentication between IPsec peers.

1.4.1.2.4 Security management

The Web Management Application of the TOE provides the capabilities for configuration and administration. The Web Management Application can be accessed via the dedicated local Ethernet port configured for “out-of-band” management. There is no local access such as a serial console port. Therefore, the local and remote management is considered the same for this evaluation.

An authorized administrator has the ability to modify, edit, and delete security parameters such as audit data, configuration data, and user authentication data. The Web Management Application also offers an authorized administrator the capability to manage how security functions behave. For example an administrator can enable/disable certain audit functions query and set encryption/decryption algorithms used for network packets.

1.4.1.2.5 Protection of the TSF

Internal testing of the TOE hardware, software, and software updates against tampering ensures that all security functions are running and available before the TOE accepts any communications. The TSF prevents reading of pre-shared keys, symmetric keys, private keys, and passwords. The TOE uses electronic signature verification before any firmware/software updates are installed.

1.4.1.2.6 TOE access

The TOE provides the following TOE Access functionality:

- TSF-initiated session termination when a connection (remote or local) is idle for a configurable time period
- Administrative termination of own session
- TOE Access Banners

1.4.1.2.7 Trusted path/channels

The TOE protects interactive communication with administrators using TLS/HTTPS, both integrity and disclosure protection is ensured.

The TOE uses IPsec to protect communication with network entities, such as a log server, NTP server and LDAP server. This prevents unintended disclosure or modification of logs and management information.

1.4.2 TOE Documentation

- Ultra Electronics 3eTI 636-Series User’s Guide, August 15, 2017, 290000533-002, Revision E (**Admin Guide**)

2. Conformance Claims

This TOE is conformant to the following CC specifications:

- Common Criteria for Information Technology Security Evaluation Part 2: Security functional components, Version 3.1, Revision 4, September 2012.
    - Part 2 Extended
- Common Criteria for Information Technology Security Evaluation Part 3: Security assurance components, Version 3.1 Revision 4, September 2012.
    - Part 3 Conformant
- Package Claims:
    - collaborative Protection Profile for Network Devices, Version 1.0, 27 February 2015 (NDcPP10)
- Technical Decisions Applied in this ST:
    - TD0090
    - TD0116
    - TD0125
    - TD0130
    - TD0154
    - TD0156
    - TD0160
    - TD0224

2.1 Conformance Rationale

The ST conforms to the NDcPP10. As explained previously, the security problem definition, security objectives, and security requirements have been drawn from the PP.

3. Security Objectives

The Security Problem Definition may be found in the NDcPP10 and this section reproduces only the corresponding Security Objectives for operational environment for reader convenience. The NDcPP10 offers additional information about the identified security objectives, but that has not been reproduced here and the NDcPP10 should be consulted if there is interest in that material.

In general, the NDcPP10 has defined Security Objectives appropriate for network devices and as such are applicable to the CyberFence 3e-636 Series Network Security Devices TOE.

3.1 Security Objectives for the Operational Environment

OE.ADMIN_CREDENTIALS_SECURE The administrator's credentials (private key) used to access the TOE must be protected on any other platform on which they reside.

OE.NO_GENERAL_PURPOSE There are no general-purpose computing capabilities (e.g., compilers or user applications) available on the TOE, other than those services necessary for the operation, administration and support of the TOE.

OE.NO_THRU_TRAFFIC_PROTECTION The TOE does not provide any protection of traffic that traverses it. It is assumed that protection of this traffic will be covered by other security and assurance measures in the operational environment.

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 guidance documentation in a trusted manner.

OE.UPDATES The TOE firmware and software is updated by an administrator on a regular basis in response to the release of product updates due to known vulnerabilities.

4. Extended Components Definition

All of the extended requirements in this ST have been drawn from the NDcPP10. The NDcPP10 defines the following extended requirements and since they are not redefined in this ST the NDcPP10 should be consulted for more information in regard to those CC extensions.

Extended SFRs:

- FAU_STG_EXT.1: Protected Audit Event Storage
- FCS_HTTPS_EXT.1: HTTPS Protocol
- FCS_IPSEC_EXT.1: IPsec Protocol
- FCS_RBG_EXT.1: Random Bit Generation
- FCS_TLSS_EXT.1: TLS Server Protocol
- FIA_PMG_EXT.1: Password Management
- FIA_UAU_EXT.2: Password-based Authentication Mechanism
- FIA_UIA_EXT.1: User Identification and Authentication
- FIA_X509_EXT.1: X.509 Certificate Validation
- FIA_X509_EXT.2: X.509 Certificate Authentication
- FIA_X509_EXT.3: X.509 Certificate Requests
- FPT_APW_EXT.1: Protection of Administrator Passwords
- FPT_SKP_EXT.1: Protection of TSF Data (for reading of all symmetric keys)
- FPT_TST_EXT.1: TSF testing
- FPT_TUD_EXT.1: Trusted update
- FTA_SSL_EXT.1: TSF-initiated Session Locking

5. Security Requirements

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

The SFRs have all been drawn from the NDcPP10. The refinements and operations already performed in the NDcPP10 are not identified (e.g., highlighted) here, rather the requirements have been copied from the NDcPP10 and any residual operations have been completed herein. Of particular note, the NDcPP10 made a number of refinements and completed some of the SFR operations defined in the Common Criteria (CC) and that PP should be consulted to identify those changes if necessary.

The SARs are also drawn from the NDcPP10 which includes all the SARs for EAL 1. However, the SARs are effectively refined since requirement-specific 'Assurance Activities' are defined in the NDcPP10 that serve to ensure corresponding evaluations will yield more practical and consistent assurance than the EAL 1 assurance requirements alone. The NDcPP10 should be consulted for the assurance activity definitions.

5.1 TOE Security Functional Requirements

The following table identifies the SFRs that are satisfied by CyberFence 3e-636 Series Network Security Devices TOE.

Requirement Class	Requirement Component
FAU: Security audit	FAU_GEN.1: Audit Data Generation
	FAU_GEN.2: User identity association
	FAU_STG_EXT.1: Protected Audit Event Storage
FCS: Cryptographic support	FCS_CKM.1: Cryptographic Key Generation
	FCS_CKM.2: Cryptographic Key Establishment
	FCS_CKM.4: Cryptographic Key Destruction
	FCS_COP.1(1): Cryptographic Operation (AES Data Encryption/Decryption)
	FCS_COP.1(2): Cryptographic Operation (Signature Generation and Verification)
	FCS_COP.1(3): Cryptographic Operation (Hash Algorithm)
	FCS_COP.1(4): Cryptographic Operation (Keyed Hash Algorithm)
	FCS_HTTPS_EXT.1: HTTPS Protocol
	FCS_IPSEC_EXT.1: IPsec Protocol
	FCS_RBG_EXT.1: Random Bit Generation
	FCS_TLSS_EXT.1: TLS Server Protocol
FIA: Identification and authentication	FIA_PMG_EXT.1: Password Management
	FIA_UAU.7: Protected Authentication Feedback
	FIA_UAU_EXT.2: Password-based Authentication Mechanism
	FIA_UIA_EXT.1: User Identification and Authentication
	FIA_X509_EXT.1: X.509 Certificate Validation
	FIA_X509_EXT.2: X.509 Certificate Authentication
	FIA_X509_EXT.3: X.509 Certificate Requests
FMT: Security management	FMT_MOF.1(1): Management of security functions behavior - Trusted Update
	FMT_MTD.1(1): Management of TSF Data
	FMT_SMF.1: Specification of Management Functions
	FMT_SMR.2: Restrictions on Security Roles
FPT: Protection of the TSF	FPT_APW_EXT.1: Protection of Administrator Passwords
	FPT_SKP_EXT.1: Protection of TSF Data (for reading of all symmetric keys)

	FPT_STM.1: Reliable Time Stamps
	FPT_TST_EXT.1: TSF testing
	FPT_TUD_EXT.1: Trusted update
FTA: TOE access	FTA_SSL.3: TSF-initiated Termination
	FTA_SSL.4: User-initiated Termination
	FTA_SSL_EXT.1: TSF-initiated Session Locking
	FTA_TAB.1: Default TOE Access Banners
FTP: Trusted path/channels	FTP_ITC.1: Inter-TSF trusted channel
	FTP_TRP.1: Trusted Path

Table 1 TOE Security Functional Components

5.1.1 Security audit (FAU)

5.1.1.1 Audit Data Generation (FAU_GEN.1)

FAU_GEN.1.1

The TSF shall be able to generate an audit record of 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; and
c) All administrative actions comprising:
- Administrative login and logout (name of user account shall be logged if individual user accounts are required for administrators).
- Security related configuration changes (in addition to the information that a change occurred it shall be logged what has been changed).
- Generating/import of, changing, or deleting of cryptographic keys (in addition to the action itself a unique key name or key reference shall be logged).
- Resetting passwords (name of related user account shall be logged).
- Starting and stopping services (if applicable).
- [***no other actions***];

d) Specifically defined auditable events listed in Table 1.

Requirement	Auditable Events	Additional Content
FAU_GEN.1		
FAU_GEN.2		
FAU_STG_EXT.1		
FCS_CKM.1		
FCS_CKM.2		
FCS_CKM.4		
FCS_COP.1(1)		
FCS_COP.1(2)		
FCS_COP.1(3)		
FCS_COP.1(4)		
FCS_HTTPS_EXT.1	Failure to establish a HTTPS Session.	Reason for failure.
FCS_IPSEC_EXT.1	Failure to establish an IPsec SA.	Reason for failure.
FCS_RBG_EXT.1		
FCS_TLSS_EXT.1	Failure to establish a TLS Session.	Reason for failure.
FIA_PMG_EXT.1		
FIA_UAU.7		
FIA_UAU_EXT.2	All use of identification and authentication mechanism.	Origin of the attempt (e.g., IP address).

FIA_UIA_EXT.1	All use of identification and authentication mechanism.	Provided user identity, origin of the attempt (e.g., IP address).
FIA_X509_EXT.1	Unsuccessful attempt to validate a certificate.	Reason for failure.
FIA_X509_EXT.2		
FIA_X509_EXT.3		
FMT_MOF.1(1)	Any attempt to initiate a manual update.	
FMT_MTD.1(1)	All management activities of TSF data.	
FMT_SMF.1		
FMT_SMR.2		
FPT_APW_EXT.1		
FPT_SKP_EXT.1		
FPT_STM.1	Changes to time.	The old and new values for the time. Origin of the attempt to change time for success and failure (e.g., IP address).
FPT_TST_EXT.1		
FPT_TUD_EXT.1	Initiation of update; result of the update attempt (success or failure).	
FTA_SSL.3	The termination of a remote session by the session locking mechanism.	
FTA_SSL.4	The termination of an interactive session.	
FTA_SSL_EXT.1	Any attempts at unlocking of an interactive session.	
FTA_TAB.1		
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.
FTP_TRP.1	Initiation of the trusted path. Termination of the trusted path. Failure of the trusted path functions.	Identification of the claimed user identity.

FAU_GEN.1.2

The TSF shall record within each audit record at least the following information:
a) Date and time of the event, type of event, subject identity, and the outcome (success or failure) of the event; and
b) 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 1.

5.1.1.2 User identity association (FAU_GEN.2)

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.

5.1.1.3 Protected Audit Event Storage (FAU_STG_EXT.1)

FAU_STG_EXT.1.1

The TSF shall be able to transmit the generated audit data to an external IT entity using a trusted channel according to FTP_ITC.1.

FAU_STG_EXT.1.2

The TSF shall be able to store generated audit data on the TOE itself.

FAU_STG_EXT.1.3

The TSF shall [***overwrite previous audit records according to the following rule: [when allotted space has reached its threshold]***] when the local storage space for audit data is full.

5.1.2 Cryptographic support (FCS)

5.1.2.1 Cryptographic Key Generation (FCS_CKM.1)

FCS_CKM.1.1

The TSF shall generate asymmetric cryptographic keys in accordance with a specified cryptographic key generation algorithm: [
- RSA schemes using cryptographic key sizes of 2048-bit or greater that meet the following: FIPS PUB 186-4, 'Digital Signature Standard (DSS)', Appendix B.3,
- ECC schemes using 'NIST curves' [P-256, P-384, P-521] that meet the following: FIPS PUB 186-4, 'Digital Signature Standard (DSS)', Appendix B.4,
- FFC schemes using cryptographic key sizes of 2048-bit or greater that meet the following: FIPS PUB 186-4, 'Digital Signature Standard (DSS)', Appendix B.1].

5.1.2.2 Cryptographic Key Establishment (FCS_CKM.2)

FCS_CKM.2.1

The TSF shall perform cryptographic key establishment in accordance with a specified cryptographic key establishment method: [
- RSA-based key establishment schemes that meets the following: NIST Special Publication 800-56B, 'Recommendation for Pair-Wise Key Establishment Schemes Using Integer Factorization Cryptography',
- Elliptic curve-based key establishment schemes that meets the following: NIST Special Publication 800-56A, 'Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography',
- Finite field-based key establishment schemes that meets the following: NIST Special Publication 800-56A, 'Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography'].

5.1.2.3 Cryptographic Key Destruction (FCS_CKM.4)

FCS_CKM.4.1

The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method

- For plaintext keys in volatile storage, the destruction shall be executed by a [***single overwrite consisting of [zeroes]***];
- For plaintext keys in non-volatile storage, the destruction shall be executed by the invocation of an interface provided by a part of the TSF that [***o logically addresses the storage location of the key and performs a [[single] -pass] overwrite consisting of [a new value of the key]***]

that meets the following: No Standard. (TD0130 applied)

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

FCS_COP.1(1).1

The TSF shall perform encryption/decryption in accordance with a specified cryptographic algorithm AES used in [***CBC, GCM***] mode and cryptographic key sizes [***128 bits, 256 bits***] that meet the following: AES as specified in ISO 18033-3, [***CBC as specified in ISO 10116, GCM as specified in ISO 19772***].

5.1.2.5 Cryptographic Operation (Signature Generation and Verification) (FCS_COP.1(2))

FCS_COP.1(2).1

The TSF shall perform cryptographic signature services (generation and verification) in accordance with a specified cryptographic algorithm [
- RSA Digital Signature Algorithm and cryptographic key sizes (modulus) [2048 bits],
- Elliptic Curve Digital Signature Algorithm and cryptographic key sizes [256 bits]] that meet

the following: [***- For RSA schemes: FIPS PUB 186-4, 'Digital Signature Standard (DSS)', Section 5.5, using PKCS #1 v2.1 Signature Schemes RSASSA-PSS and/or RSASSAPKCS1v1_5, ISO/IEC 9796-2, Digital signature scheme 2 or Digital Signature scheme 3 (TD0116 applied), - For ECDSA schemes: FIPS PUB 186-4, 'Digital Signature Standard (DSS)', Section 6 and Appendix D, Implementing 'NIST curves' P-256, P-384, and [P-521] , ISO/IEC 14888-3, Section 6.4***].

5.1.2.6 Cryptographic Operation (Hash Algorithm) (FCS_COP.1(3))

FCS_COP.1(3).1

The TSF shall perform cryptographic hashing services in accordance with a specified cryptographic algorithm [***SHA-1, SHA-256, SHA-384, SHA-512***] that meet the following: ISO/IEC 10118-3:2004.

5.1.2.7 Cryptographic Operation (Keyed Hash Algorithm) (FCS_COP.1(4))

FCS_COP.1(4).1

The TSF shall perform keyed-hash message authentication in accordance with a specified cryptographic algorithm [***HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512***] and cryptographic key sizes [***160, 256, 384, 512***] and message digest sizes [***160, 256, 384, 512***] bits that meet the following: ISO/IEC 9797-2:2011, Section 7 'MAC Algorithm 2'.

5.1.2.8 HTTPS Protocol (FCS_HTTPS_EXT.1)

FCS_HTTPS_EXT.1.1

The TSF shall implement the HTTPS protocol that complies with RFC 2818.

FCS_HTTPS_EXT.1.2

The TSF shall implement HTTPS using TLS.

FCS_HTTPS_EXT.1.3

The TSF shall establish the connection only if [***the peer initiates handshake***]. (TD0125 applied)

5.1.2.9 IPsec Protocol (FCS_IPSEC_EXT.1)

FCS_IPSEC_EXT.1.1

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

FCS_IPSEC_EXT.1.2

The TSF shall have a nominal, final entry in the SPD that matches anything that is otherwise unmatched, and discards it.

FCS_IPSEC_EXT.1.3

The TSF shall implement [***transport mode, tunnel mode***]. (TD0160 applied)

FCS_IPSEC_EXT.1.4

The TSF shall implement the IPsec protocol ESP as defined by RFC 4303 using the cryptographic algorithms AES-CBC-128, AES-CBC-256 (both specified by RFC 3602) and [***AES-GCM-128 (specified in RFC 4106), AES-GCM-256 (specified in RFC 4106)***] together with a Secure Hash Algorithm (SHA)-based HMAC.

FCS_IPSEC_EXT.1.5

The TSF shall implement the protocol: [***- IKEv2 as defined in RFC 5996 and [with mandatory support for NAT traversal as specified in RFC 5996, section 2.23], and [no other RFCs for hash functions]***].

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 3602 and [***AES-GCM-128, AES-GCM-256 as specified in RFC 5282***].

FCS_IPSEC_EXT.1.7

The TSF shall ensure that [***- IKEv2 SA lifetimes can be configured by an Security Administrator based on [o length of time, where the time values can configured within [24] hours]***].

FCS_IPSEC_EXT.1.8

The TSF shall ensure that [***- IKEv2 Child SA lifetimes can be configured by a Security Administrator based on [o number of bytes,***
o length of time, where the time values can be configured within [8] hours]].

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 [***112 (for DH Group 14), 128 (for group 19), 192 (for group 20)***] bits.

FCS_IPSEC_EXT.1.10

The TSF shall generate nonces used in [***IKEv2***] exchanges of length [***- [112 (for DH Group 14), 128 (for group 19), 192 (for group 20)]***].

FCS_IPSEC_EXT.1.11

The TSF shall ensure that IKE protocols implement DH Groups(s) [***14 (2048-bit MODP), 19 (256-bit Random ECP), 20 (384-bit Random ECP)***]. (TD0224 applied)

FCS_IPSEC_EXT.1.12

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.

FCS_IPSEC_EXT.1.13

The TSF shall ensure that all IKE protocols perform peer authentication using [***RSA, ECDSA***] that use X.509v3 certificates that conform to RFC 4945 and [***Pre-shared Keys***].

FCS_IPSEC_EXT.1.14

The TSF shall only establish a trusted channel to peers with valid certificates.

5.1.2.10 Random Bit Generation (FCS_RBG_EXT.1)

FCS_RBG_EXT.1.1

The TSF shall perform all deterministic random bit generation services in accordance with ISO/IEC 18031:2011 using [***CTR_DRBG (AES)***].

FCS_RBG_EXT.1.2

The deterministic RBG shall be seeded by at least one entropy source that accumulates entropy from [***[a single] hardware-based noise source***] with a minimum of [***256 bits***] of entropy at least equal to the greatest security strength, according to ISO/IEC 18031:2011 Table C.1 'Security Strength Table for Hash Functions', of the keys and hashes that it will generate.

5.1.2.11 TLS Server Protocol (FCS_TLSS_EXT.1)

FCS_TLSS_EXT.1.1

The TSF shall implement [***TLS 1.2 (RFC 5246), TLS 1.1 (RFC 4346)***] supporting the following ciphersuites:

Mandatory Ciphersuites:
- TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268

Optional Ciphersuites:

[***- TLS_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268,***
- TLS_DHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268,
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268,
- TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246,
- TLS_RSA_WITH_AES_256_CBC_ SHA256 as defined in RFC 5246,
- TLS_DHE_RSA_WITH_AES_128_CBC_ SHA256 as defined in RFC 5246,
- TLS_DHE_RSA_WITH_AES_256_CBC_ SHA256 as defined in RFC 5246].

FCS_TLSS_EXT.1.2

The TSF shall deny connections from clients requesting SSL 2.0, SSL 3.0, TLS 1.0, and [***none***]. (TD0156 applied)

FCS_TLSS_EXT.1.3

The TSF shall generate key establishment parameters using RSA with key size 2048 bits and [***no other size***] and [***Diffie-Hellman parameters of size 2048 bits and [no other size]***].

5.1.3 Identification and authentication (FIA)

5.1.3.1 Password Management (FIA_PMG_EXT.1)

FIA_PMG_EXT.1.1

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

a) Passwords shall be able to be composed of any combination of upper and lower case letters, numbers, and the following special characters: [***'!', '@', '#', '$', '%', '^', '&', '*', '(', ')', ['+'. '-', '_']***] ;

b) Minimum password length shall be settable by the Security Administrator, and shall support passwords of 15 characters or greater.

5.1.3.2 Protected Authentication Feedback (FIA_UAU.7)

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.

5.1.3.3 Password-based Authentication Mechanism (FIA_UAU_EXT.2)

FIA_UAU_EXT.2.1

The TSF shall provide a local password-based authentication mechanism, [***[remote password-based authentication mechanism]***] to perform administrative user authentication.

5.1.3.4 User Identification and Authentication (FIA_UIA_EXT.1)

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; - [***allow network packets configured by the administrator to flow through the TOE***]

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.

5.1.3.5 X.509 Certificate Validation (FIA_X509_EXT.1)

FIA_X509_EXT.1.1

The TSF shall validate certificates in accordance with the following rules:

- RFC 5280 certificate validation and certificate path validation.
- The certificate path must terminate with a trusted CA certificate.
- The TSF shall validate a certificate path by ensuring the presence of the basicConstraints extension and that the CA flag is set to TRUE for all CA certificates.
- The TSF shall validate the revocation status of the certificate using [***a Certificate Revocation List (CRL) as specified in RFC 5280 Section 6.3***].
- The TSF shall validate the extendedKeyUsage field according to the following rules:
    - Certificates used for trusted updates and executable code integrity verification shall have the Code Signing purpose (id-kp 3 with OID 1.3.6.1.5.5.7.3.3) in the extendedKeyUsage field.
    - Server certificates presented for TLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field.

- Client certificates presented for TLS shall have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the extendedKeyUsage field.

FIA_X509_EXT.1.2

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

5.1.3.6 X.509 Certificate Authentication (FIA_X509_EXT.2)

FIA_X509_EXT.2.1

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

FIA_X509_EXT.2.2

When the TSF cannot establish a connection to determine the validity of a certificate, the TSF shall [***accept the certificate***].

5.1.3.7 X.509 Certificate Requests (FIA_X509_EXT.3)

FIA_X509_EXT.3.1

The TSF shall generate a Certificate Request Message as specified by RFC 2986 and be able to provide the following information in the request: public key and [***Common Name, Organization, Country***].

FIA_X509_EXT.3.2

The TSF shall validate the chain of certificates from the Root CA upon receiving the CA Certificate Response.

5.1.4 Security management (FMT)

5.1.4.1 Management of security functions behavior - Trusted Update (FMT_MOF.1(1))

FMT_MOF.1(1).1

The TSF shall restrict the ability to enable the functions to perform manual update to Security Administrators.

5.1.4.2 Management of TSF Data (FMT_MTD.1(1))

FMT_MTD.1(1).1

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

5.1.4.3 Specification of Management Functions (FMT_SMF.1)

FMT_SMF.1.1

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

- Ability to administer the TOE locally and remotely;
- Ability to configure the access banner;
- Ability to configure the session inactivity time before session termination or locking;
- Ability to update the TOE, and to verify the updates using [***digital signature***] capability prior to installing those updates (Per TD0090);
- [***o Ability to configure audit behavior,***
  o Ability to configure the cryptographic functionality].

5.1.4.4 Restrictions on Security Roles (FMT_SMR.2)

FMT_SMR.2.1

The TSF shall maintain the roles: - Security 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 - The Security Administrator role shall be able to administer the TOE locally; - The Security Administrator role shall be able to administer the TOE remotely are satisfied.

5.1.5 Protection of the TSF (FPT)

5.1.5.1 Protection of Administrator Passwords (FPT_APW_EXT.1)

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

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

FPT_SKP_EXT.1.1

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

5.1.5.3 Reliable Time Stamps (FPT_STM.1)

FPT_STM.1.1

The TSF shall be able to provide reliable time stamps.

5.1.5.4 TSF testing (FPT_TST_EXT.1)

FPT_TST_EXT.1.1

The TSF shall run a suite of the following self-tests [***during initial start-up (on power on)***] to demonstrate the correct operation of the TSF: [***Software integrity test, AES Known Answer Test, SHA Known Answer Test, HMAC Known Answer Test, DRBG Known Answer Test, RSA Known Answer Test, ECDSA Known Answer Test,***].

5.1.5.5 Trusted update (FPT_TUD_EXT.1)

FPT_TUD_EXT.1.1

The TSF shall provide Security Administrators the ability to query the currently executing version of the TOE firmware/software and [***no other TOE firmware/software version***]. (TD0154 applied)

FPT_TUD_EXT.1.2

The TSF shall provide Security Administrators the ability to manually initiate updates to TOE firmware/software and [***no other update mechanism***].

FPT_TUD_EXT.1.3

The TSF shall provide means to authenticate firmware/software updates to the TOE using a [***digital signature mechanism***] prior to installing those updates.

5.1.6 TOE access (FTA)

5.1.6.1 TSF-initiated Termination (FTA_SSL.3)

FTA_SSL.3.1

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

5.1.6.2 User-initiated Termination (FTA_SSL.4)

FTA_SSL.4.1

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

5.1.6.3 TSF-initiated Session Locking (FTA_SSL_EXT.1)

FTA_SSL_EXT.1.1

The TSF shall, for local interactive sessions, [***- terminate the session***] after a Security Administrator-specified time period of inactivity.

5.1.6.4 Default TOE Access Banners (FTA_TAB.1)

FTA_TAB.1.1

Refinement: 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.

5.1.7 Trusted path/channels (FTP)

5.1.7.1 Inter-TSF trusted channel (FTP_ITC.1)

FTP_ITC.1.1

The TSF shall be capable of using [***IPsec***] to provide a trusted communication channel between itself and authorized IT entities supporting the following capabilities: audit server, [***[accessing an NTP server]***] that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from disclosure and detection of modification of the channel data.

FTP_ITC.1.2

The 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 [***audit logging, and accessing an NTP server***].

5.1.7.2 Trusted Path (FTP_TRP.1)

FTP_TRP.1.1

The TSF shall be capable of using [***TLS, HTTPS***] to provide a communication path between itself and authorized 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 provides detection of modification of the channel 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.

5.2 TOE Security Assurance Requirements

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

Requirement Class	Requirement Component
ADV: Development	ADV_FSP.1: Basic functional specification

AGD: Guidance documents	AGD_OPE.1: Operational user guidance
	AGD_PRE.1: Preparative procedures
ALC: Life-cycle support	ALC_CMC.1: Labelling of the TOE
	ALC_CMS.1: TOE CM coverage
ASE: Security Target	ASE_TSS.1: Security Target
ATE: Tests	ATE_IND.1: Independent testing - conformance
AVA: Vulnerability assessment	AVA_VAN.1: Vulnerability survey

Table 2 Assurance Components

5.2.1 Development (ADV)

5.2.1.1 Basic functional specification (ADV_FSP.1)

ADV_FSP.1.1d
The developer shall provide a functional specification.

ADV_FSP.1.2d
The developer shall provide a tracing from the functional specification to the SFRs.

ADV_FSP.1.1c
The functional specification shall describe the purpose and method of use for each SFR-enforcing and SFR-supporting TSFI.

ADV_FSP.1.2c
The functional specification shall identify all parameters associated with each SFR-enforcing and SFR-supporting TSFI.

ADV_FSP.1.3c
The functional specification shall provide rationale for the implicit categorisation of interfaces as SFR-non-interfering.

ADV_FSP.1.4c
The tracing shall demonstrate that the SFRs trace to TSFIs in the functional specification.

ADV_FSP.1.1e
The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADV_FSP.1.2e
The evaluator shall determine that the functional specification is an accurate and complete instantiation of the SFRs.

5.2.2 Guidance documents (AGD)

5.2.2.1 Operational user guidance (AGD_OPE.1)

AGD_OPE.1.1d
The developer shall provide operational user guidance.

AGD_OPE.1.1c
The operational user guidance shall describe, for each user role, the user-accessible functions and privileges that should be controlled in a secure processing environment, including appropriate warnings.

AGD_OPE.1.2c
The operational user guidance shall describe, for each user role, how to use the available interfaces provided by the TOE in a secure manner.

AGD_OPE.1.3c
The operational user guidance shall describe, for each user role, the available functions and interfaces, in particular all security parameters under the control of the user, indicating secure values as appropriate.

AGD_OPE.1.4c

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

AGD_OPE.1.5c

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

AGD_OPE.1.6c

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

AGD_OPE.1.7c

The operational user guidance shall be clear and reasonable.

AGD_OPE.1.1e

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

5.2.2.2 Preparative procedures (AGD_PRE.1)

AGD_PRE.1.1d

The developer shall provide the TOE including its preparative procedures.

AGD_PRE.1.1c

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

AGD_PRE.1.2c

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

AGD_PRE.1.1e

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

AGD_PRE.1.2e

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

5.2.3 Life-cycle support (ALC)

5.2.3.1 Labelling of the TOE (ALC_CMC.1)

ALC_CMC.1.1d

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

ALC_CMC.1.1c

The TOE shall be labelled with its unique reference.

ALC_CMC.1.1e

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

5.2.3.2 TOE CM coverage (ALC_CMS.1)

ALC_CMS.1.1d

The developer shall provide a configuration list for the TOE.

ALC_CMS.1.1c

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

ALC_CMS.1.2c

The configuration list shall uniquely identify the configuration items.

ALC_CMS.1.1e

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

5.2.4 Security Target (ASE)

5.2.4.1 Security Target (ASE_TSS.1)

ASE_TSS.1.1c

Refinement: The TOE summary specification shall describe how the TOE meets each SFR. In the case of entropy analysis the TSS is used in conjunction with required supplementary information on Entropy.

5.2.5 Tests (ATE)

5.2.5.1 Independent testing - conformance (ATE_IND.1)

ATE_IND.1.1d

The developer shall provide the TOE for testing.

ATE_IND.1.1c

The TOE shall be suitable for testing.

ATE_IND.1.1e

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

ATE_IND.1.2e

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

5.2.6 Vulnerability assessment (AVA)

5.2.6.1 Vulnerability survey (AVA_VAN.1)

AVA_VAN.1.1d

The developer shall provide the TOE for testing.

AVA_VAN.1.1c

The TOE shall be suitable for testing.

AVA_VAN.1.1e

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

AVA_VAN.1.2e

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

AVA_VAN.1.3e

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

6. TOE Summary Specification

This chapter describes the security functions:

- Security audit
- Cryptographic support
- Identification and authentication
- Security management
- Protection of the TSF
- TOE access
- Trusted path/channels

6.1 Security audit

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

FAU_GEN.1:
The TOE is designed to be able to generate log records for a wide range of security relevant and other events as they occur. The events that can cause an audit record to be logged include starting and stopping the audit function, Security Administrator's configuration of CSPs and security functions as well as all of the events identified in **Error! Reference source not found.**. The TOE generates records for several separate classes of events: authentication/access to the system, actions taken directly on the system by network clients, and management of security functions by authorized administrators.
All audit records include the date/time of the event, the identity associated with the event (such as the service, computer or user), the success/failure of the event and a definition of the event (by code or explanation).

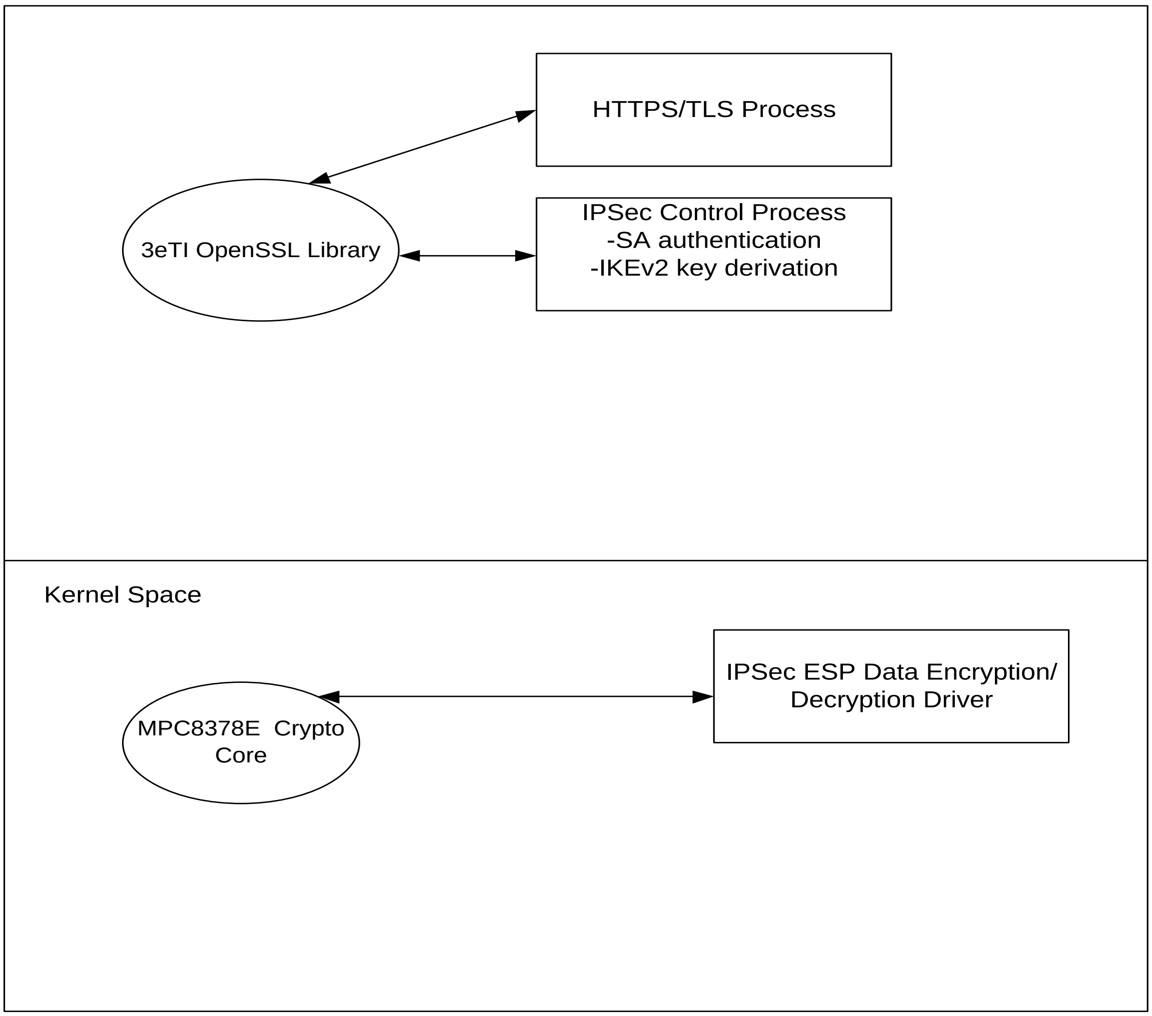
FAU_GEN.2:
All actions performed by the TOE are associated with a unique identifier, this information is maintained in the audit record, allowing the events stored there to be traced directly to the user or system for which they were performed.

FAU_STG_EXT.1:
The TOE stores audit logs locally with up to a fixed size of 256K bytes. The Security Administrator can configure the TOE to send email alert upon the audit logs reaching a configurable percentage of the fixed size.
Local password based authentication and authorization limits the access to the local audit log records. Only the Security Administrator can gain access to the local audit log records.
When the TOE is configured to export audit logs to an external SYSLOG server, it simultaneously sends the message to the server and local store. The TOE requires the external audit server and itself to be connected via an IPsec session. The User's Guide provides details about the "Export Audit Logs" configuration.
The TOE exports audit data over IPsec using AES128/256 bit encryption. Disconnection to external entities such as syslog server will result in log of communication error and attempt to re-establish secure channel. At no point, will plaintext be transmitted. The TOE does not implement an automatic synchronization mechanism between the local and remote audit storage.
When the audit log storage space is full, the TOE also provides the Authorized Administrator the option of overwriting "old" audit records rather than preventing auditable events.

6.2 Cryptographic support

There are two cryptographic engines within the TOE models as shown in the figure below. All models share the same hardware MPC8378E cryptographic core and OpenSSL library.

Figure 6-1: TOE Cryptographic Cores



First is the 3eTI's own OpenSSL library. 3eTI's OpenSSL Library serves as the sole user application level cryptographic library. It provides the FCS_COP functions listed below. All user level applications, such as HTTPS/TLS Web UI, IPsec SA authentication module use this library. 3eTI's OpenSSL provides the following cryptographic algorithms in FIPS mode: AES, RSA, HMAC, SHS, ECDSA, DH, DRBG

The 3eTI OpenSSL Library represents not the entire OpenSSL Library, but the FIPS Object Module that is compiled into the larger OpenSSL Library. Because 3eTI has already compiled the FIPS Object Module and then links that same, identical Object Module into different versions of the larger OpenSSL library, the version of the larger OpenSSL Library is not relevant.

There is a FreeScale MPC8378E cryptographic core within the TOE as well. It provides cryptographic function for the Linux kernel drivers. IPsec ESP data encryption/decryption using AES-CBC with SHS or AES-GCM is provided by this engine. The MPC8378E cryptographic core provides the following cryptographic algorithms in FIPS mode: AES (CBC, GCM). HMAC , SHS.

The TOE utilizes version 2.0 of its OpenSSL Algorithm Implementation and version 1.0 of the MPC8378E cryptographic core in version 5.1.300 of the TOE.

Table 6-1: TOE FIPS-140 Tested Algorithms

Algorithm	Cert No.	SFR Mapping
3eTI OpenSSL		
AES (CBC, 128, 256 bits key)	2060	FCS_COP.1(1)
CVL KAS FFC/ECC	1357	FCS_CKM.2(1)
DSA, KeyPairGen	1255	FCS_CKM.1(1)
ECDSA PKG/PKV/SigGen/SigVer P256/384/521	415 303	FCS_COP.1(2) FCS_CKM.1(1)
SHS	1801	FCS_COP.1(3)
HMAC	1253	FCS_COP.1(4)
RSA key generation	2568	FCS_CKM.1(1)
RSA sign/verify	1491	FCS_COP.1(2)
DRBG NIST SP800-90 with one independent hardware based noise source of 256 bits of non-deterministic	822	FCS_RBG_EXT.1
MPC8378E Cryptographic Core		
AES (CBC)	2078	FCS_COP.1(1)
AES (GCM)	2105	FCS_COP.1(1)
HMAC	1259	FCS_COP.1(4)
SHS	1807	FCS_COP.1(3)

FCS_CKM.1:

The TOE support both RSA and ECDSA key generation. The key generation is used by the TOE when it creates a Certificate Signing Request (CSR) to apply for a certificate from the Certificate Authority (CA). The TOE enforces the key size of 2048 or greater for RSA key pairs and supports NIST curves P256, P384 and P512 for ECDSA key pairs.

FCS_CKM.2

The TOE acts as both receiver and sender for RSA-based key establishment and ECDSA based key establishment in cryptographic operations.

FCS_CKM.4: Table 6- below lists all the keys and CSPs used and managed by the TOE.

Table 6-2: TOE CSPs Use and Management

Non-Protocol Keys/CSPs						
Key/CSP	**Type**	**Generation/ Input**	**Output**	**Storage**	**Zeroization**	**Use**
Operator passwords	ASCII string	Input encrypted (using TLS session key)	Not output	PKCS5 hash in flash	Zeroized when reset to factory settings.	Used to authenticate Security Admin and Admin role operators
Firmware verification key	ECDSA public key	Embedded in firmware at compile time.	Not output	Plaintext in flash	Zeroized when firmware is	Used for firmware digital

		Firmware upgrade is through encrypted (using TLS session key)			upgraded.	signature verification
RBG Keys/CSPs						
Key/CSP	**Type**	**Generation/ Input**	**Output**	**Storage**	**Zeroization**	**Use**
DRBG CTR V	32-byte value	32 bytes from /dev/random file, /dev/random is populated by hardware noise generator	Not output	Plaintext in RAM	Zeroized every time a new random number is generated using the FIPS DRBG after it is used.	Used as CTR V value for FIPS DRBG.
DRBG CTR Key	32-byte value	32 bytes from /dev/random file, /dev/random is populated by hardware noise generator	Not output	Plaintext in RAM	Zeroized every time a new random number is generated using the FIPS DRBG after it is used.	Used as CTR key for FIPS DRBG.
RFC 2818 HTTPS Keys/CSPs						
Key/CSP	**Type**	**Generation/ Input**	**Output**	**Storage**	**Zeroization**	**Use**
RSA private key	RSA (2048) (key wrapping; key establishment methodology provides 112-bits of encryption strength)	Not input (installed at factory)	Not output	Plaintext in flash	Zeroized when new private key is uploaded	Used to support Security Admin and Admin TLS/HTTPS interfaces.
TLS session key for encryption	AES (128/256)	Not input, derived using TLS protocol	Not output	Plaintext in RAM	Zeroized when a page of the web GUI is served after it is used.	Used to protect TLS/HTTPS session.
Public Security Parameter						
HTTPS Public certificate	RSA (2048)	Input encrypted (using TLS session key)	During TLS session setup			Used to setup TLS session for TLS/HTTPS
HTTPS root certificate	RSA (2048)	Input encrypted (using TLS session key)	Not output			Used to setup TLS session for TLS/HTTPS
IPsec Keys						
DH Private Key	2048 bits private key	Generated	Not output	Plaintext in RAM	Zeroized when no longer used	IKE v2 SA setup

ECCDH Private Key	256,384,521 bits	Generated	Not output	Plaintext in RAM and encrypted in FLASH	RAM copy zeroized when no longer used	IKE v2 SA setup
IPSec SA Authenticatio n certificate private key	RSA (2048, 4096), ECDSA (256,384,512)	Input encrypted using TLS session key	Not output	Plaintext in RAM and encrypted in FLASH	RAM copy zeroized when no longer used	IKE v2 SA authentication
IPSec SA private key password	Text string	Input encrypted using TLS session key	Not output	Plaintext in RAM and encrypted in FLASH	Zeroized when no longer used	Encrypt the IPSec SA certificate private key
IPSec SA session key	Derived from DH/ECCDH key exchange	Not input	Not output	Plaintext in RAM	Zeroized when no longer used	Encrypt and Authenticate SA_Auth messages of IKE v2
IPSec ESP symmetric Data encryption key	AES, AES_GCM (128, 256)	Not input (derived from SA setup)	Not output	Plaintext in RAM	Zeroized when child SA lifetime expired	Encrypt IPSec ESP data

The zeroization technique is to write all 0xa5, then 0x5a, 0xff and finally all zeros to the memory location where the key is stored. A read-verify is performed after the zeroization. The same zeroization technique is applied to flash. The TOE does not store keys in EEPROM.

FCS_COP.1(1):

AES is implemented with key sizes of 128, and 256 bits in Cipher Block Chaining (CBC) mode and Galois Counter Mode (GCM).
The 3eTI's OpenSSL Library provides AES services for application level data encryption and decryption. The management interface uses this library to provide Transport Layer Security (TLS/HTTPS). For TOE's TLS interface, AES_CBC with 128 or 256 bits key is used.
3eTI's MPC8378E Cryptographic Core provides AES_GCM and AES_CBC services for IPsec data encryption. 128 and 256 bits keys are supported. Table 6-1 lists the AES mode and key sizes, all AES algorithm implementations are NIST CAVP validated.

FCS_COP.1(2):

The 3eTI OpenSSL Library provides the RSA Digital Signature Algorithm (rDSA) to the TLS/HTTPS Daemon for the TLS session. The TLS/HTTPS Daemon enforces a 2048 or larger bits RSA key length for use with the RSA. TOE Firmware's digital signature is using ECDSA with P256. The 3eTI OpenSSL library provides ECDSA sign/verify operation support. IPsec tunnels can be configured to use RSA (2048, 4096) or ECDSA with key size 256, 384, and 521 bits using NIST curve P256, P384 and P521) certificate for IPsec SA authentication. Table 6-1 lists RSA and ECDSA CAVP validation certificate numbers.

FCS_COP.1(3):
The TSF supports SHA-1, SHA-256, SHA-384, and SHA-512 for secure hashing. See Table 6-1 for details.
The security hashing functions are used in IPsec IKEv2 and ESP to provide data packet integrity.

FCS_COP.1(4):

The TOE's OpenSSL Library and the MPC8378E cryptographic core both implement HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512, keyed-hash message authentication supporting digest sizes:

160, 256, 384, and 512 bits and key size of 160 bits, 256 bits, 384 and 512 bits implemented to meet ISO/IEC 9797-2:2011, Section 7 “MAC Algorithm 2

FCS_IPSEC_EXT.1:
The TOE implements IPsec protocol in full compliance with IETF RFCs as specified by NDcPP. Within the TOE, NTP client uses IPsec tunnel with NTP server and audit log service will use IPsec tunnel to the remote log server.

The TOE supports IKEv2 only as defined by RFCs 5996 and always attempts NAT traversal, hence an administrator need not configure either. During the Security Association (SA) setup phase, the TOE supports the following DH groups:

- ecp384
- ecp256
- modp2048

If the administrator selects “auto negotiation” from IPsec “Cipher Suites” configuration GUI, then the groups listed above will be send to the IPsec peer during the IKEv2 negotiation. If “Suite B GCM128” is selected, then the TOE will use ecp256 group. If “Suite B GCM256” is selected, then the TOE will use ecp384 group.

The TOE choses and enforces the group and AES cipher to make sure that the SA confidentiality strength is equivalent or greater than the configured ESP confidentiality strength. For example, the TOE (when configured for “Auto Negotiation”) will reject any ESP proposal with an AES key length greater than the negotiated IKE AES key length. Similarly, administrator selection of “Suite B GCM128” or similar ciphers ensure the TOE will enforce a single, proscribed set of modes to make sure that the parents IPsec SA confidentiality strength is equal or greater than the child SA’s strength.

Mode	**IKE (openssl library)**				**ESP (Hardware encryption)**	
	Encryption	**Integrity**	**Pseudo Random Function**	**DH Group**	**Encryption**	**Integrity (where applicable)**
Suite B GCM 128	aes128cbc	sha256	sha256	ecp256	aes128gcm128	-

The TOE uses ISO/IEC 18031:2011 DRBG to generate the “x” in each DH group and the nonce. After the Diffie Hellman exchanges that setup the session keys, the IKEv2 payload is protected by the following encryption algorithms:

- AES-CBC-256
- AES-CBC-128

SHA-512, SHA-384, SHA-256 and SHA1 are used to provide payload data integrity. X.509 certificates with rDSA 2048 bits or larger key or ECDSA 256, 384, and 521 bits key with NIST P256, P384 and P521 are used for IPsec tunnel authentication with its peer.

The TOE supports IPsec tunnel mode and transport mode which allows on the payload of packet to be encrypted. The TOE requires no administrative configuration and negotiates either depending on the peer. The TOE uses IPsec standard encapsulating Security Payload (ESP) protocol to provide authentication, encryption and anti-replay services. It uses the following ciphers to encrypt the IPsec data payload:

1. GCM mode with Nonce length of 128, 96 and 64 bits
    - AES-GCM-128
    - AES-GCM-256
2. CBC mode with HMAC-SHA-512, HMAC-SHA-384, HMAC-SHA-256 and HMAC-SHA1 as integrity
    - AES-CBC-128
    - AES-CBC-256

When the administrator chooses AES-GCM-128 for ESP encryption, the TOE will automatically choose SHA-256 for the IKE's integrity and pseudo random function. If the choice is AES-GCM-256, then SHA-256 will be used in IKE integrity and SHA-384 will be used for the pseudo random function. If the administrator chooses "Auto Negotiation" for IPsec, the TOE will send cipher list ranked with the highest security first to its peer. For example, the IKE integrity list will be sent as: SHA-512, SHA-384, SHA-256 and SHA1. It's expected that the peer will pick the strongest one it could support. There is no need to explicitly configure security hashing functions in the IPsec configuration.

The IPsec daemon module implements implicit policies such that only expected data packages are allowed. Any data packages that violate the policy will be discarded.

The TOE allows the Security Administrator to configure the IKEv2 SA and child SA lifetime by minutes (20-1440, default 180) and the TOE additionally allows the Administrator to configure child SA lifetime by number of bytes (90 to 2047Kb, 0 for unlimited) or by number of data packets (192 to 2097151K, 0 for unlimited).

The TOE supports X.509 certificate for IPsec mutual authentication. RSA certificate with 2048 or 4098 bits, ECDSA certificates with 256, 384, and 521 bits key is supported implementing NIST curves P256, P384 and P521. When certificates are used for authentication, the distinguished name (DN) is verified to ensure the certificate is valid and is from a valid entity. The DN naming attributes in the certificate is compared with the expected DN naming attributes (as specified by an administrator) and deemed valid if the attribute types are the same and the values are the same and as expected.

The TOE can be configured to use pre-shared key for IPsec authentication as well. The security administrator first select "Pre-Shared Key" under the "Authentication" when configure the IPsec tunnel via web GUI, then enter the pre-shared key manually via the GUI.

The TOE's IPsec Security Policy Database (SPD) is dynamically configured based the trusted paths IPsec is used to protect. For example, the TOE uses IPsec to protect a remote syslog trusted path. In this instance, records are written into the SPD to protect packets passing between the TOE and the remote syslog server based on source address, destination address, protocol and port number. When protecting remote syslog trusted path, the SPD will have record matching ingress UDP traffic with source address and port corresponding to the remote syslog server. Additionally, the SPD will have record matching egress traffic with destination address and port corresponding to the remote syslog server. Traffic passing through the security boundary and matching either of these two records will be classified as "PROTECTED" using IPsec transport mode. Traffic that does not match any records in the SPD but does match the local firewall access list will be allows to "BYPASS" the security boundary unperturbed. Traffic that does not match any records in the SPD and doesn't match the local firewall access list will be "DISCARDED". Additional records are written into the SPD when additional trusted paths are configured for IPsec protection (i.e. remote audit log and NTP server).

FCS_RBG_EXT.1:
The TOE implements RBG as defined ISO/IEC 18031:2011 using AES. The entropy source is a hardware based noise generator. Entropy is obtained from a zener diode operated in avalanche mode. The noise from the diode is passed through a cascaded pair of operational amplifiers, then applied to the input of a Microchip MCP3221. MCP3221 is a successive approximation analog to digital converter (ADC) with a 12 bit resolution.
The TOE communicates with the MCP3221 hardware over the 2-wire I2C and reads in the raw entropy. The raw entropy is further conditioned by the Linux kernel to produce 8 bits of entropy per byte. Then the random bytes are read by the DRBG implementation of 256 bits of DRBG key and DRBG seed.

FCS_HTTPS_EXT.1:

FCS_TLSS_EXT.1:
The management interface with remote administration station is always TLS/HTTPS. The HTTPS implementation is fully compliant with RFC 2818. The TOE acts as an HTTPS server and waits for client connections on TCP port

443. The TOE's HTTPS server permits an HTTP client to close the connection at any time, and the HTTPS server will recover gracefully. In particular, the HTTPS server is prepared to receive an incomplete close from the client, and is willing to resume TLS sessions closed in this fashion.
The TOE's HTTPS server supports TLS version 1.1/1.2 only and will deny connection requests from TLS clients with lower version. It supports the following ciphers:

- TLS_RSA_WITH_AES_128_CBC_SHA
- TLS_RSA_WITH_AES_256_CBC_SHA
- TLS_DHE_RSA_WITH_AES_128_CBC_SHA
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA
- TLS_RSA_WITH_AES_128_CBC_SHA256
- TLS_RSA_WITH_AES_256_CBC_SHA256
- TLS_DHE_RSA_WITH_AES_128_CBC_SHA256
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA256

The TOE's TLS/HTTPS server uses RSA 2048 bits certificate for TLS authentication. After the TLS session's successful setup, the security administrator logs into the TOE via user name and passwords. If the failure count reaches the configured threshold, the TLS/HTTPS session will be terminated by the TLS/HTTPS server. The Diffie-Hellman group 14 with parameters of size 2048 bits is used for the key agreement.

6.3 Identification and authentication

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

FIA_PMG_EXT.1:
The TOE supports the local definition of users with corresponding passwords. The passwords can be composed of any combination of upper and lower case letters, numbers, and special characters (that include: "!", "@", "#", "$", "%", "^", "&", "*", "(", and ")". Minimum password length is settable by the Authorized Administrator, and can be configured for minimum password lengths of 15 characters. Additionally, the TOE supports password lengths up to 32 characters long. NOTE: The TOE will truncate passwords that are longer than 32 characters when creating a user or changing passwords for an existing user.

FIA_UAU.7: when a user is entering their password information, the password is obscured such that no observer could read the password off the screen.

FIA_UAU_EXT.2:

FIA_UIA_EXT.1:
The TOE requires all users to be successfully identified and authenticated before allowing any TSF mediated actions to be performed except for the login banner that is displayed prior to user authentication and any network packets as configured by the authorized administrator may flow through the TOE. Network packets as configured by the authorized administrator may flow through the TOE.

The administrator logs on the TOE through either dedicated local Ethernet port or over WAN Ethernet port to access the Web Management UI. The Web Management UI is accessible over HTTPS only and the TOE supports TLS 1.1/1.2. There is no local access such as a serial console port. Therefore, the local and remote management is considered the same for this evaluation.
A successful authentication is determined by a successful username and password combination after the HTTPS connection. An incorrect password will result in a failed authentication attempt. The TOE does not provide a reason for failure in the cases of a login failure.

The TOE supports local authentication using A local user database.

FIA_X509_EXT.1:

FIA_X509_EXT.2:

FIA_X509_EXT.3:

The TOE uses X.509 certificates for IPsec authentications. The TOE can be configured with the certificates and their corresponding private key by security administrator or by creating CSRs and importing the CA signed CSRs to the TOE. The security administrator can load and delete certificates for usage of IPsec authentication, load and delete CAs, intermediated CAs and CRLs. The TOE checks that the basicConstraints extension and CA flag are set to TRUE for all CA certificates before their acceptance. During the IPsec authentication using X509 certificate, the TOE develops a certificate path from a trust anchor configured by security administrator which is fully compliant with RFC 5280. When a certificate chain is received from a peer, the TOE processes the certificate chain path until the first trusted certificate, or trust point, is reached. The TOE uses CRL that have been preloaded by the administrator to validate the peer certificates, and the TOE will accept as valid, a certificate for which the administrator has not loaded a CRL. Failure by the TOE to establish the certification path to a trust anchor will lead to the failure of establishment of this IPsec trusted channel.

The TOE allows the security administrator to view the certificates and CAs. The TOE's GUI will display the certificates' name, subject name, issuer name, valid-start date, expiration date.

6.4 Security management

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

FMT_MOF.1(1):

FMT_MTD.1(1):

FMT_SMF.1:

FMT_SMR.2:

The Web Management Application over HTTP/TLS provides capabilities for the authorized administrator to manage cryptographic, audit, and authentication functions and data.

The TOE provides three roles: Security Administrator, Non-security Administrator, and user (Peer Device). Security Administrator and Administrator can only access the TOE through Web Application through TLS/HTTPS.
Upon successful authentication with the TOE, the Security Administrator can manage TSF data as shown in the table below.

Table 6-2: Management of TSF Data

Service and Purpose	Details	Security Administrator (referred to as Crypto officer in guidance)	Non-Security Administrator
Input of Keys	IKE v2 digital certificate private key, 802.1X supplicant private key, device HTTPS private keys, authentication key with RADIUS Server.	X	
Create and manage users	Support up to 10 administrator users and 5 crypto officer users.	X	
Change password	Administrator changes his own password only.	X	X
Show system status	View traffic status and systems log excluding security audit log.	X	X
Manage audit logging	Select audit events to be logged. Configure remote audit logging. View audit event records.	X	
Key zeroization via reboot		X	X

Factory default	Delete all configurations and set device back to factory default state.	X	
Perform Self-Test	Run algorithm KAT through reboot.	X	X
Load New Firmware	Upload 3eTI digitally signed firmware.	X	
SNMP Management	Manage all SNMP settings including SNMPv3 encryption key.	X	X
HTTPS Management	Load HTTPS server certificate and private key.	X	
Key Generation	Create asymmetric key pairs and X509v3 Certificate Signing Request.	X	X

No GUI interfaces are accessible to the user prior to authentication. The TOE enforces authentication then enables the TSF data configuration interfaces. The Non-security administrators have no access to those TSF data configuration interfaces.

6.5 Protection of the TSF

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

FPT_APW_EXT.1:

FPT_SKP_EXT.1:
The authentication passwords are stored in PKCS5 format in the TOE. All other CSPs are stored in encrypted format in the TOE on non-volatile memory. The file system that holds the hashed password and encrypted CSPs are made read-only during runtime to avoid data corruption. None of the files or CSPs is available through any external interfaces to users/administrators. The Web Application Interface allows security administrator to input keys/passwords to the TOE with no output capabilities.

FPT_STM.1:
The TOE has a running NTP daemon to synchronize the local time with an external NTP server. IPsec tunnel is setup between the TOE and NTP server to protect the integrity and privacy of the time source. In the absence of an NTP server in the Operational Environment, the authorized administrator has the capability to set the time locally. The local time is used for the following security functions identified in this ST:

- Time stamping each audit record.
- Verifying the validity of the Web Server X509v3 Certificate.
- Verifying the validity of the IPsec tunnel peer's Certificate.
- Verifying the validity of the Firmware X509v3 Certificate during the firmware upload process.
- Enforcing user lockout periods for "Bad Password" login attempts.
- Timing out login sessions due to inactivity.

FPT_TST_EXT.1:

The TSF performs a firmware integrity check and a configuration file integrity check on system start up. Algorithm Known Answer Tests are run at startup time as shown below:

Power-on self-tests:
Software Integrity Test

- Bootloader Integrity Test
- Firmware Integrity Test

FreeScale PowerQUICC Crypto Engine Power-on self-tests:

- AES_GCM encrypt/decrypt KAT
- SHA-1, SHA256, SHA384, SHA512 KAT
- HMAC SHA-1, SHA256, SHA384, SHA512 KAT

3eTI OpenSSL library Power-on self-tests:

• HMAC SHA-1, SHA256, SHA384, SHA512	KAT
• SHA-1, SHA256, SHA384, SHA512	KAT
• FIPS SP800-90 DRBG	KAT
• RSA sign/verify	KAT
• ECDSA sign/verify	KAT

Vectors for each known answer test (KAT) are compiled into the Firmware. The known inputs are provided to the cryptographic function and the output of that function is compared to the known output. The firmware is halted if any of the known answer tests fail.
After device is powered on, the first thing done by bootloader is to check its own integrity. If the integrity is broken, firmware won't boot. Firmware integrity is performed at firmware boot up. Both firmware and bootloader are digitally signed with ECDSA.

The TOE also performs the DRBG Continuous Random Number test consists of a Repetitive Count test and an Adaptive Proportion test. Each random sample is compared to previous samples. The Repetitive Count test ensures the new sample is not repeated sequentially above a threshold. The Adaptive Proportion test ensures the new sample is not repeated beyond a threshold within a window of previous samples.

FPT_TUD_EXT.1:

When newer version of firmware is released, the customers are notified by 3eTI's customer service department, normally via e-mail. If a customer desires to get a copy of the new firmware, the customer will be provided with an URL link to the secured download site together with onetime valid user name and password.
The Security Administrator can update the TOE's firmware. The firmware is digitally signed with ECDSA. The TOE uses the public key to verify the digital signature. Upon successful verification, the TOE will load the new update upon reboot. The update will be rejected if the verification fails.

6.6 TOE access

The TOE access function is designed to satisfy the following security functional requirements:

FTA_SSL.3:

FTA_SSL.4:

FTA_SSL_EXT.1:

The Web Management Application terminates the remote or local session if it detects inactivity longer than the configured time period. The default time period is 10 minutes. The remote session will be closed by the Web Management Application together with the HTTPS session. The Security Administrator is required to re-authenticate with the TOE and setup a new session. The time intervals are configurable by the security administrator.

FTA_TAB.1:
The Management GUI displays a customizable TOE access banner to the remote administrative user before the user can log into the system.

6.7 Trusted path/channels

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

FTP_ITC.1:

The TOE provide a trusted communication channel between itself and all authorized IT entities that is logically distinct from other communication channels. IPsec is setup between the TOE and the audit log server, LDAP and NTP server. The trusted channel can be initiated either by the TOE or by the remote IT entities.

FTP_TRP.1:
All remote administrative communications take place over a secure encrypted TLS session. The HTTPS/TLS implementation allows web browser clients to connect to TOE HTTPS server. The remote users are able to initiate TLS communications with the TOE.