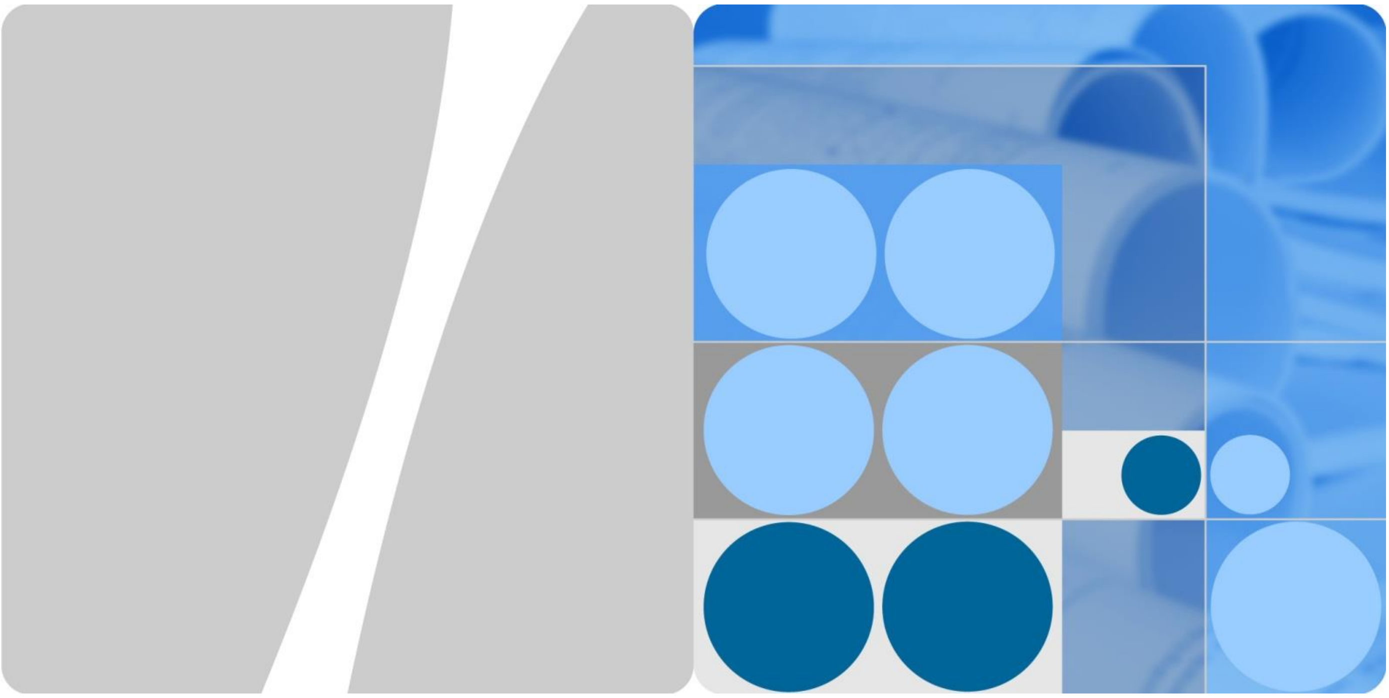


Huawei NE40E Series Software Consisting of VRP and the Underlying OS

Security Target Lite

Issue	2.5
Date	2018-10-23

HUAWEI TECHNOLOGIES CO., LTD.

Contents

1 Introduction

1.1 ST reference

Title: Security Target Lite of Huawei NE40E Series Software Consisting of VRP and the Underlying OS

Version: V2.5

Author/Developer: Huawei Technologies Co., Ltd.

Certification ID: BSI-DSZ-CC-1053

Subject: Router, High-end network product

Keywords: Security Target, Common Criteria, Huawei, NE40E, VRP, Router

1.2 TOE reference

TOE name: Huawei NE40E Series Software Consisting of VRP and the Underlying OS

TOE software version: Base on the V800R010C00SPC200, patch the V800R010SPH220T which is released on August 21, 2018

Running on Hardware Models: NE40E-X3A and NE40E-X16A

1.3 TOE overview

The NE40E series routers include NE40E-X3A and NE40E-X16A, satisfying the requirements for networks of various scales. The NE40E is a high-end network product developed by Huawei. It is deployed at the edge of IP backbone networks, IP metropolitan area networks (MANs), and other large-scale IP networks. It consists of both hardware and software, providing network traffic processing capacity. The TOE is software only and is consisting of the Versatile Routing Platform (VRP) and the underlying Operating System (OS). Network traffic is processed and forwarded by the underlying hardware according to routing decisions downloaded from VRP.

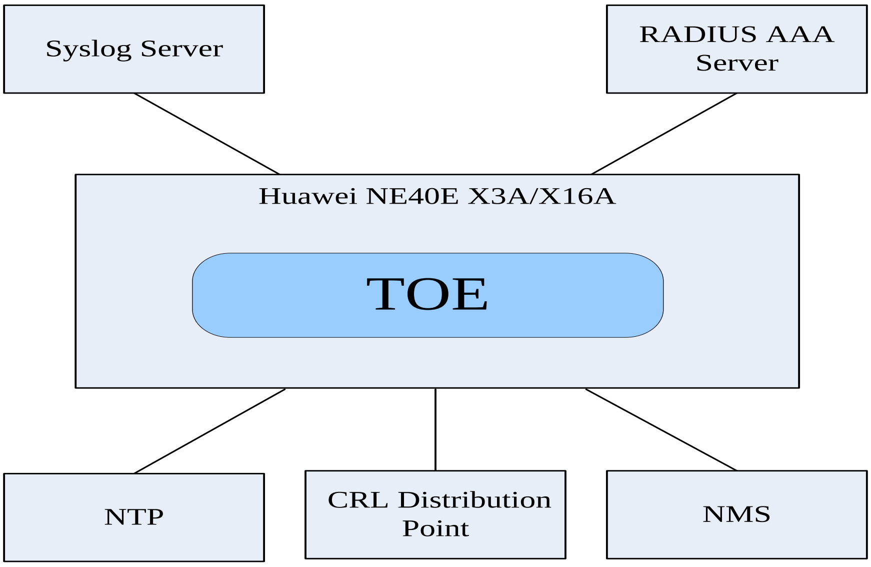
The NE40E runs with the VRP developed by Huawei. VRP provides extensive security features. These features include different interfaces with according access levels for administrators, enforcing authentications prior to establishment of administrative sessions, auditing of security-relevant management activities. These security features constitute the TOE.

1.3.1 TOE usage

1. The TOE supports username/password, or public-key authentication mode and only users that are authenticated can access the TOE and its command line interface.
2. The TOE is accessed by CLI locally or a Network Management Server (NMS) remotely over SSH so that a secure channel is established to protect the data between TOE and NMS.
3. For secure transmission of audit information between the TOE and the Syslog server a secure TLS channel is used.
4. The TOE supports digital signature verification for software. Each of the software package or patch package released by Huawei includes a unique digital signature. When an NMS distributes the package to NE40E, the TOE will verify the online digital signature before updating. The verification of the digital signature demonstrates the integrity and authenticity of the package. The package is only processed further after successful verification of the digital signature, otherwise the package will be discarded without processing.

The TOE provides security services onto a single and secure device. It supports (in some cases optionally) the following hardware, software, and firmware in its environment when the TOE is configured in Figure 1-1 (NTP: Network Time Protocol; NMS: Network Management Server).

Figure 1-1 IT Entities which connect with TOE



These IT entities (like the NTP server) should be physical protected in order to ensure that no one can attack them or stole information.

1.3.2 TOE type

The TOE type is a network device that is connected to the network and has an infrastructure role within the network.

1.3.3 Non TOE Hardware and Software

The TOE supports the following hardware, software, and firmware components in its operational environment. Each component is identified as being "Required" or "Optional" based on the claims made in this Security Target. All of the following environment components are supported by all TOE evaluated configurations.

Table 1-1 IT Environment Components

Component	Required/ Optional	Usage/Purpose Description for TOE performance
NE40E-X3A, NE40E-X16A	Required	The TOE runs on these hardware platforms.
RADIUS AAA Server	Optional	This RADIUS AAA server provides user authentication. The TOE correctly leverages the services provided by this RADIUS AAA server to provide authentication to administrators.
Network Management Server	Required	This includes any Management workstation with a SSH client installed that is used to establish a protected channel with the TOE.
Local Console	Required	This includes any Console that is directly connected to the TOE via the Serial Console Port and is used by the TOE administrator to support TOE administration.
CRL Distribution Point	Optional	CRL should be downloaded from CRL Distribution Point. Then this downloaded CRL should be uploaded into the TOE.
NTP	Optional	The TOE supports secure communications with an NTP server in order to synchronize the date and time on the TOE with the NTP server's date and time. When the TOE acts as NTP server, it receives NTP request from client and send timestamp to the client.
Syslog Server	Optional	This includes any syslog server to which the TOE would transmit syslog messages.

1.4 TOE description

1.4.1 Architectural overview

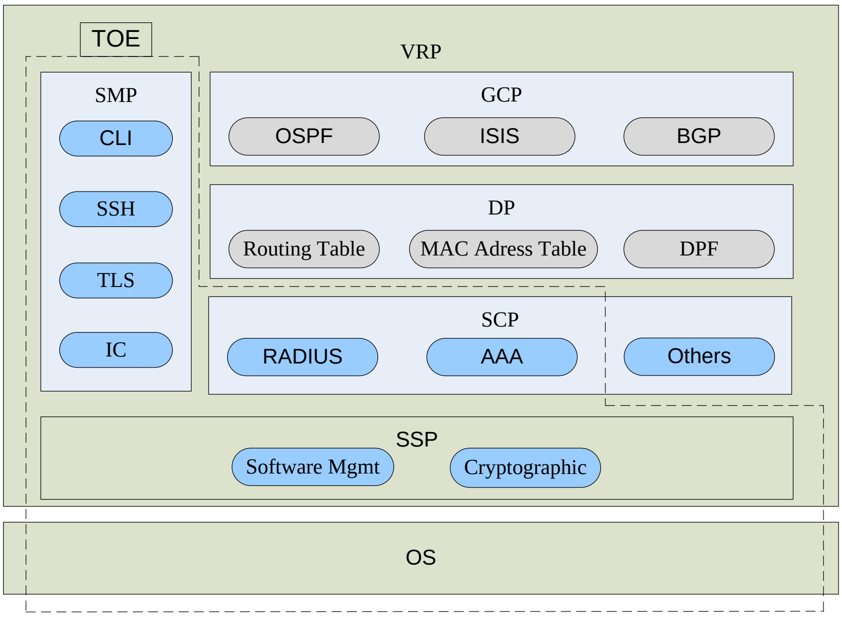
This section will introduce TOE from a software architectural view. The TOE is composed of two parts: VRP and the underlying OS.

The underlying OS is Linux. The OS provides basic services including memory management, scheduling management, file management, and device management.

The VRP is a new-generation network operating platform, which has a distributed, multi-process, and component-based architecture. It builds upon the hardware development trend and will meet carriers' exploding service requirements for the next five to ten years.

The VRP software is responsible for functional management, routing information generation, receiving generated routing information and formatting them into hardware-specific data to direct traffic forwarding.

Figure 1-2 TOE Software Architecture



Remark: VRP consists of SMP, SCP, GCP, DP and SSP. SMP, SCP, SSP and OS are in the scope of the TOE, GCP and DP are not in the scope of the TOE

6 logical planes are defined for the Software Architecture, they are:

(1) System Manage Plane(SMP), implements management for external access, management for system configuration, information output on VRP;

(2) Service Control Plane(SCP), implements authentication, authorization, accounting and other serviceable functionality on VRP;

(3) General Control Plane(GCP), implements routing information learning, ARP table entry learning, STP(Spanning Tree Protocol) topology management, and functionalities related to TCP/IP stack on VRP;

(4) System Service Plane (SSP), implements system internal scheduling, communication, management of signals, events, timers, etc. Communication with Virtual Path is also

implemented at this plane.

(5) Data Plane (DP), implements traffic forwarding. Forwarding related information, e.g. routing information, ARP table entry, static MAC table entry are generated in GCP and downloaded via communication channel provided by SSP.

(6) Operating System (OS), provide hardware and software resource management. The underlying OS is the Windriver Linux and the Real Time Operating System (RTOS). The Real Time Operating System (RTOS) manages underlying hardware resources, such as the CPU, memory, storage devices, network devices, and other hardware. The RTOS is developed by Huawei.

SMP, SCP, GCP are hosted in the MPU, DP are hosted in the LPU. SSP and OS are hosted in the both MPU and LPUs.

Table 1-2 Modules specifications

Modules	**Evaluated/ Not-Evaluated**	**Description**
AAA	Evaluated	AAA (Authentication Authorization Accounting), implemented in accordance with related RFC, provides authentication, authorization and accounting functionalities.
BGP	Not-Evaluated	Border Gateway Protocol, the protocol backing the core routing decisions on the Internet. It maintains a table of IP networks or 'prefixes' which designate network reachability among autonomous systems (AS). It is described as a path vector protocol.
CLI	Evaluated	Command line interface component, a basic component deployed in VRP. It implements management of commands registration, command parsing, and privileging.
Cryptographic	Evaluated	RSA, SHA, HMAC-SHA, AES, DRBG
DPF	Not-Evaluated	The DPF (Data Packet Forwarding) provides interfaces that send and receive packets on a router, while processing the packets at a high speed and switching data packets inside the router.
IC	Evaluated	Information Center, accepts, categorizes and filters information generated by all components and/or modules including log and alarm information, and outputs accordingly (e.g., to terminal, to log file).
ISIS	Not-Evaluated	Intermediate System to Intermediate System (IS-IS) is a dynamic routing protocol initially designed by the International Organization for Standardization (ISO) for its Connectionless Network Protocol (CLNP). To support IP routing, the Internet Engineering Task Force (IETF) extends and modifies IS-IS in RFC 1195, which enables IS-IS to be applied to both TCP/IP and Open System Interconnection (OSI) environments. This type of IS-IS is called Integrated IS-IS or Dual IS-IS.

MAC Address Table	Not-Evaluated	Each device maintains a MAC address table. A MAC address table stores MAC addresses, VLAN IDs, and outbound interfaces learned from other devices, listed in Table 1. To forward data, the device searches the MAC address table to quickly locate the outbound interface based on the destination MAC address and VLAN ID in the data frame. This implementation reduces broadcast traffic.
OSPF	Not-Evaluated	Open Shortest Path First, is an adaptive routing protocol for Internet Protocol (IP) networks. It uses a link state routing algorithm and falls into the group of interior routing protocols, operating within a single autonomous system (AS).
Radius	Evaluated	As one of the commonly-used protocols that implement Authentication, Authorization and Accounting (AAA), RADIUS was initially used to manage a large number of geographically-dispersed users that use serial ports and modems. The Radius here is client which will communicate with server.
Routing Table	Not-Evaluated	A router searches a routing table for routes, and each router maintains at least one routing table. Routing tables store the routes discovered by various routing protocols.
Software Management	Evaluated	You can select a proper operation to upgrade and maintain the device according to the real-world situation. Application scenarios of these operations are as follows: • System software upgrade System software upgrade can optimize device performance, add new features, and upgrade the current software version. • Patch installation Patches are a type of software compatible with system software. They are used to fix urgent bugs in system software. You can upgrade the system by installing patches, without having to upgrade the system software.
SSH	Evaluated	Secure Shell, provides secure channel between end user and the TOE, and to protect the TOE from IP address fraud, password interception, etc.
TLS	Evaluated	TLS function performs loading digital certificate revocation list, and trusted CA file.
Others	Not-Evaluated	Other non-TSF functionalities which are not within the scope of this evaluation.

1.4.2 Physical scope

TOE is software only (software package version : V800R010C00SPC200, HPV: V800R010SPH220T) and is running on NE40E X3A and NE40E X16A devices. The hardware is out of the TOE scope. The product provides several HW models. These models differ in their modularity and throughput by supplying more slots in hosting chassis, but they offer exchangeable forwarding unit modules, switch fabrics, and use the same version of VRP software, i.e. the same TOE version. Each device contains two MPUs of the same type for redundancy.

The model ID (e.g. X3A) is built according to the following scheme:

- X as first character denotes ‘eXtend’ which means that for this device the number of LPUs can be extended.
- A as last character denotes ‘Advanced’ which means that the device has higher forwarding performance compared to the former ('non-A') models like NE40E X3.
- The number in between the first and last character denotes the maximum quantity of LPUs that can be inserted in this specific chassis.

The two models can be equipped with the following MPUs:

- NE40E X3A MPU revision: Main Processing Unit D4 (Abbreviated as MPUD4) and Main Processing Unit D4(16G Memory) (Abbreviated as MPUD4(16G Memory))
- NE40E X16A MPU revision: Main Processing Unit B6 (Abbreviated as MPUB6)

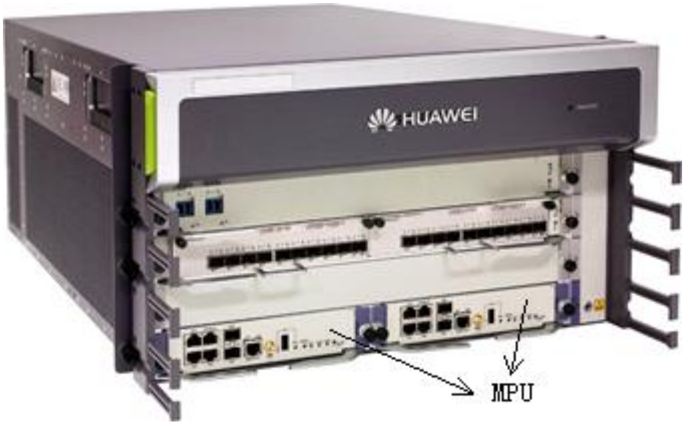
NE40E X3A differs from NE40E X16A in the following ways:

Item	**NE40E-X16A (MPUB6)**	**NE40E-X3A (MPUD4, 16G)**	**NE40E-X3A (MPUD4)**
Switching capacity	50.32 Tbps	2.76 Tbps	2.76 Tbps
Forwarding performance	11520 Mpps	900 Mpps	900 Mpps
Number of LPUs	16 LPUs	3 LPUs	3 LPUs
Processing unit	Octa-core 2.3GHZ	Quad-core 2.0GHz	Quad-core 2.0GHz
SDRAM	32 GB	8 GB x 2	8 GB x 1
Flash	16 MB	16 MB	16 MB
Storage	SSD card: 32GB	SSD card:8 GB	SSD card:8 GB
Operating System	RTOS	WindRiver Linux	WindRiver Linux
CPU	X86	PowerPC	PowerPC

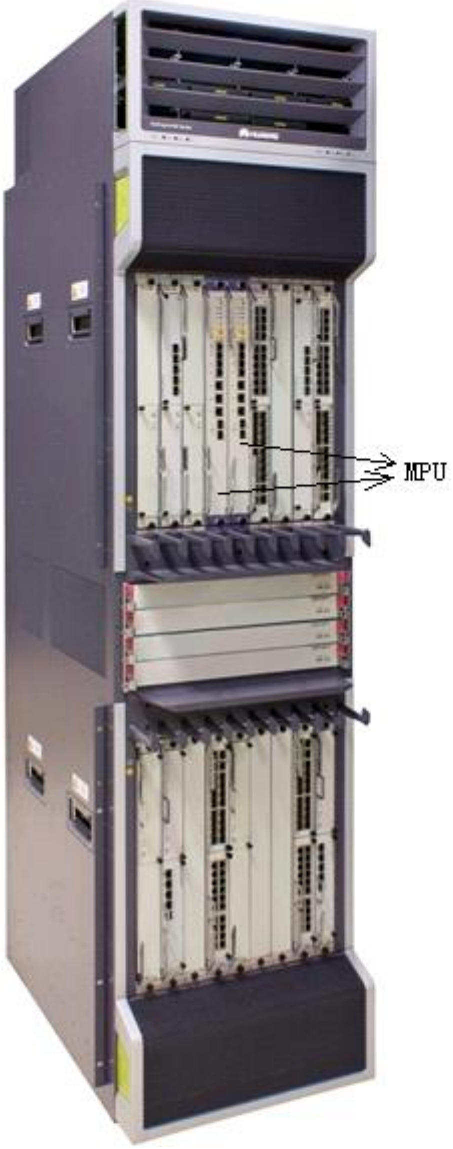
Table 1-3 Details all physical units on which the TOE runs

Boards	Supported Interfaces and Usage
MPU/SRU	As the system control and management unit, the MPU provides the following functions on the system control panel: (1) Route calculation: All routing protocol packets are sent by the forwarding engine to the MPU for processing. In addition, the MPU broadcasts and filters packets, and downloads routing policies from the policy server. (2) Outband communication between boards: The LAN switch modules integrated on the MPU provide outband communications between boards. In this manner, messages can be controlled, maintained, and exchanged between SFUs and LPUs. (3) Device management and maintenance: Devices can be managed and maintained through the management interfaces (serial interfaces) provided by the MPU. (4) Data configuration: The MPU stores configuration data, startup files, charging information, upgrade software, and system logs.
LPU	Interfaces supported by LPU are listed as below. More details about these interfaces can be found in user manual [PD], Description>Hardware Description>Boards. (1) ETH interface, connector type RJ45, operation mode 10M/100M/1000M Base-TX auto-sensing, supporting half-duplex and full-duplex, used for receiving and transmitting network traffic. (2) FE interface, connector type LC/PC optical connector, compliant to SFP optical module 100M-FX, supporting full-duplex, used for receiving and transmitting network traffic. (3) GE interface, connector type LC/PC optical connector, compliant to SFP optical module 1000Base-X-SFP, supporting full-duplex, used for receiving and transmitting network traffic. (4) 10GE interface, connector type LC/PC optical connector, compliant to XFP optical module 10GBase LAN/WAN-XFP, supporting full-duplex, used for receiving and transmitting network traffic

1) Pictures of NE40E X3A
   - NE40E-X3A



2) Pictures of NE40E X16A
   - NE40E-X16A



The TOE software available:

Users can login the HUAWEI support website to download the software packet in accordance to the version of the TOE. Users can verify the software by digital signature (The digital signature is also published on HUAWEI support website)

The guidance documents included as part of the TOE are:

(1) [PD]
(2) [PRE]
(3) [OPE]
(4) [C&R]

1.4.3 Logical Scope of the TOE

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

(1) Security audit

The log module of the host software records operations on a device and events that occur to a device. The recorded operations and events are log messages. Log messages provide evidence for diagnosing and maintaining a system. Log messages reflect the operating status of a device and are used to analyze the conditions of a network and to find out the causes of network failure or faults.

Key elements of log messages include timestamp, host name, Huawei identity, version, module name, severity, brief description, etc.

IC component are the module processing, outputting log records. Information hierarchy is designed to help the user roughly differentiate between information about normal operation and information about faults. Since the information center needs to output information to the terminal, console, log buffer, and log file.

(2) Cryptographic support

The TOE provides cryptography in support of secure connections that includes remote administrative management.

The cryptographic services provided by the TOE are described in Table below.

Table 1-4 Cryptography provided by TOE

Cryptography Function	Use in the TOE
DRBG	Used in session establishment of TLS and SSH
RSA	Used in session establishment of TLS and SSH
SHA	Used to provide cryptographic hashing services
HMAC-SHA	Used to provide integrity and authentication verification
AES	Used to encrypt traffic transmitted through TLS and SSH

(3) Identification and authentication

The authentication functionality provides validation by user’s account name and password. Public key authentication is supported for SSH users.Detailed functionalities, for example max idle-timeout period, max log-in attempts, UI lock, user kick out, can be applies by administrator according to networking environment, customized security considerations, differential user role on TOE, and/or other operational concerns.

(4) Secure Management

The TOE restricts the ability to determine the behavior of and modify the behavior of the functions transmission of audit data to the security administrator. Only the security administrator can manage the cryptographic keys. Only the security administrator has the right of opening/closing the security services and creation/deletion/modification of the user accounts.

(5) Protection of the TSF

The TOE protects the pre-shared keys, symmetric keys, and private keys from reading them by an unauthorized entity. The TOE stores the users or administrator passwords in non-plaintext form preventing them from reading. The TOE verifies the packet before their installation and uses the digital signature.

(6) TOE access through user authentication

The TOE provides communication security by implementing SSH protocol.

To protect the TOE from eavesdrop and to ensure data transmission security and confidentiality, SSH implements:

- authentication by password or by public-key;
- AES encryption algorithms;
- secure cryptographic key exchange;
- besides default TCP port 22, manually specifying a listening port is also implemented since it can effectively reduce attack.

(7) Trusted path and channels for device authentication

The TOE supports the trusted connections using TLS for the communication with the audit server.

1.4.4 Standalone TOE

[CPP_ND], chapter 3 introduces distributed TOEs, i.e. TOEs that consist of more than one component. This does not refer to different software components running on one hardware component but different software components running on separate hardware component that need to interact with each other and altogether comprise the TOE.

This ST refers to a standalone TOE which is not a distributed TOE in the sense of [CPP_ND], chapter 3. All additional requirements that are defined for distributed TOEs within [CPP_ND] are therefore ignored in this ST. There are dedicated paragraphs in several Application Notes of [CPP_ND] which are only applicable to distributed TOEs. These dedicated paragraphs have not been integrated into the Application Notes in this ST since the TOE is not a distributed TOE.

2 PP conformance claims

2.1 CC Conformance Claim

As defined by the references [CC1], [CC2] and [CC3], this ST:

- conforms to the requirements of Common Criteria v3.1, Revision 5
- is Part 2 extended, Part 3 conformant
- does not claim conformance to any other PP than the one specified in chap 2.2
- does not claim conformance to any Evaluation Assurance Level as defined in [CC3], chap. 8.

2.2 Protection Profile Conformance

This security target claims “Exact Conformance” to [CPP_ND]. Note that "Exact Conformance" is defined in [CPP_ND], chap. 2.

The methodology applied for the cPP evaluation is defined in [CEM]. In addition to [CEM], the evaluation activities for [CPP_ND] are completed in [SD_ND].

The assurance package applicable to this ST is defined in [CPP_ND] as follows:

Table 2-1 Assurance Package

Assurance Class	Assurance Components
Security Target (ASE)	Conformance claims (ASE_CCL.1)
	Extended components definition (ASE_ECD.1)
	ST introduction (ASE_INT.1)
	Security objectives for the operational environment (ASE_OBJ.1)

Assurance Class	**Assurance Components**
	Stated security requirements (ASE_REQ.1)
	Security Problem Definition (ASE_SPD.1)
	TOE summary specification (ASE_TSS.1)
Development (ADV)	Basic functional specification (ADV_FSP.1)
Guidance documents (AGD)	Operational user guidance (AGD_OPE.1)
	Preparative procedures (AGD_PRE.1)
Life cycle support (ALC)	Labeling of the TOE (ALC_CMC.1)
	TOE CM coverage (ALC_CMS.1)
Tests (ATE)	Independent testing – sample (ATE_IND.1)
Vulnerability assessment (AVA)	Vulnerability survey (AVA_VAN.1)

2.3 Conformance Rationale

2.3.1 TOE Appropriateness

The TOE provides all of the functionality at a level of security commensurate with that identified in the [CPP_ND].

2.3.2 TOE Security Problem Definition Consistency

The Threats, Assumptions, and Organization Security Policies included in the Security Target represent the Threats, Assumptions, and Organization Security Policies specified in [CPP_ND] for which conformance is claimed verbatim. All concepts covered in the collaborative Protection Profile Security Problem Definition are included in the Security Target.

2.3.3 Statement of Security Objectives Consistency

The security objectives included in the security target represent the security objectives specified in [CPP_ND] for which conformance is claimed verbatim. All concepts covered in Protection Profile`s Statement of security objectives are included in the Security Target.

2.3.4 Statement of Security Requirements Consistency

The Security Functional Requirements included in the Security Target represent the Security Functional Requirements specified in the [CPP_ND] for which conformance is claimed verbatim. All concepts covered the Protection Profile’s Statement of Security Requirements are included in the Security Target. Additionally, the Security Assurance Requirements included in the Security Target are identical to the Security Assurance Requirements included in section 6 of the [CPP_ND].

3 Security Problem Definition

3.1 Assets

The owner of the TOE presumably places value upon the following entities as long as they are in the scope of the TOE.

Table 3-1 TOE Assets

Asset Name	Description
Audit data	The data which is provided during security audit logging. TOE Security characteristic: confidentiality, integrity.
Authentication data	The data which is used to identify and authenticate the external entities such as account, password, certificate, etc. TOE Security characteristic: confidentiality, integrity.
Cryptography data	The data which is used for digital signature and encryption/decryption such as key. TOE Security characteristic: confidentiality, integrity.
Management data	The data which is used for software updates, and software integrity checking. TOE Security characteristic: integrity.
Device data	Configuration data; device firmware; software; authentication data; Cryptography data TOE Security characteristic: confidentiality (authentication data; cryptography data), integrity (all).
Critical network traffic	Administration traffic; Authentication traffic containing Authentication data; Audit traffic; traffic containing cryptography data; traffic containing Management data TOE Security characteristic: confidentiality, integrity.
Network traffic	Traffic intended for processing and forwarding ("through traffic") which is not intended for the TOE as endpoint.

Security Functionality of the Device	The TOE Security Functions (TSF) (Remark: In the context of this ST the Security Functionality of the device refers to the security functions of the TOE). TOE Security characteristic: integrity.
Network on which the device resides	The network on which the device resides. TOE Security characteristic: integrity.
Network device	The network device itself. TOE Security characteristic: integrity.
Trust relations with other network devices	Trust relations of the TOE with other network devices. TOE Security characteristic: integrity, authenticity.

3.2 Threats

The threats for the Network Device are grouped according to functional areas of the device in the sections below.

3.2.1 T.UNAUTHORIZED_ADMINISTRATOR_ACCESS

Threat agents may attempt to gain Administrator access to the network device by nefarious means such as masquerading as an Administrator to the device, masquerading as the device to an Administrator, replaying an administrative session (in its entirety, or selected portions), or performing man-in-the-middle attacks, which would provide access to the administrative session, or sessions between network devices. Successfully gaining Administrator access allows malicious actions that compromise the security functionality of the device and the network on which it resides.

SFR Rationale:

- The Administrator role is defined in FMT_SMR.2 and the relevant administration capabilities are defined in FMT_SMF.1 and FMT_MTD.1/CoreData, with additional capabilities in FMT_MOF.1/Services and FMT_MOF.1/Functions
- The actions allowed before authentication of an Administrator are constrained by FIA_UIA_EXT.1, and include the advisory notice and consent warning message displayed according to FTA_TAB.1
- The requirement for the Administrator authentication process is described in FIA_UAU_EXT.2
- Locking of Administrator sessions is ensured by FTA_SSL_EXT.1 (for local sessions), FTA_SSL.3 (for remote sessions), and FTA_SSL.4 (for all interactive sessions)
- The secure channel used for remote Administrator connections is specified in FTP_TRP.1/Admin
- (Malicious actions carried out from an Administrator session are separately addressed by T.UNDETECTED_ACTIVITY)
- (Protection of the Administrator credentials is separately addressed by T.PASSWORD_CRACKING).

3.2.2 T.WEAK_CRYPTOGRAPHY

Threat agents may exploit weak cryptographic algorithms or perform a cryptographic exhaust against the key space. Poorly chosen encryption algorithms, modes, and key sizes will allow attackers to compromise the algorithms, or brute force exhaust the key space and give them unauthorized access allowing them to read, manipulate and/or control the traffic with minimal effort.

SFR Rationale:

- Requirements for key generation and key distribution are set in FCS_CKM.1 and FCS_CKM.2 respectively
- Requirements for use of cryptographic schemes are set in FCS_COP.1/DataEncryption, FCS_COP.1/SigGen, FCS_COP.1/Hash, and FCS_COP.1/KeyedHash
- Requirements for random bit generation to support key generation and secure protocols (see SFRs resulting from T.UNTRUSTED_COMMUNICATION_CHANNELS) are set in FCS_RBG_EXT.1
- Management of cryptographic functions is specified in FMT_SMF.1

3.2.3 T.UNTRUSTED_COMMUNICATION_CHANNELS

Threat agents may attempt to target network devices that do not use standardized secure tunnelling protocols to protect the critical network traffic. Attackers may take advantage of poorly designed protocols or poor key management to successfully perform man-in-the-middle attacks, replay attacks, etc. Successful attacks will result in loss of confidentiality and integrity of the critical network traffic, and potentially could lead to a compromise of the network device itself.

SFR Rationale:

- The general use of secure protocols for identified communication channels is described at the top level in FTP_ITC.1 and FTP_TRP.1/Admin
- Requirements for the use of secure communication protocols are set for all the allowed protocols in FCS_SSHC_EXT.1, FCS_SSHS_EXT.1, FCS_TLSC_EXT.1
- Requirements for use of public key certificates to support secure protocols are defined in FIA_X509_EXT.1/Rev, FIA_X509_EXT.2, FIA_X509_EXT.3

3.2.4 T.WEAK_AUTHENTICATION_ENDPOINTS

Threat agents may take advantage of secure protocols that use weak methods to authenticate the endpoints – e.g. a shared password that is guessable or transported as plaintext. The consequences are the same as a poorly designed protocol, the attacker could masquerade as the Administrator or another device, and the attacker could insert themselves into the network stream and perform a man-in-the-middle attack. The result is the critical network traffic is exposed and there could be a loss of confidentiality and integrity, and potentially the network device itself could be compromised.

SFR Rationale:

- The use of appropriate secure protocols to provide authentication of endpoints (as in the SFRs addressing T.UNTRUSTED_COMMUNICATION_CHANNELS) are ensured by the requirements in FTP_ITC.1 and FTP_TRP.1/Admin

3.2.5 T.UPDATE_COMPROMISE

Threat agents may attempt to provide a compromised update of the software or firmware which undermines the security functionality of the device. Non-validated updates or updates validated using non-secure or weak cryptography leave the update firmware vulnerable to surreptitious alteration.

SFR Rationale:

- Requirements for protection of updates are set in FPT_TUD_EXT.1
- Certificate-based protection of signatures is specified in FPT_TUD_EXT.2, supported by the X.509 certificate processing requirements in FIA_X509_EXT.1/Rev, FIA_X509_EXT.2 and FIA_X509_EXT.3
- Requirements for management of updates are defined in FMT_SMF.1 and (for manual updates) in FMT_MOF.1/ManualUpdate

3.2.6 T.UNDETECTED_ACTIVITY

Threat agents may attempt to access, change, and/or modify the security functionality of the network device without Administrator awareness. This could result in the attacker finding an avenue (e.g., misconfiguration, flaw in the product) to compromise the device and the Administrator would have no knowledge that the device has been compromised.

SFR Rationale:

- Requirements for basic auditing capabilities are specified in FAU_GEN.1 and FAU_GEN.2, with timestamps provided according to FPT_STM_EXT.1
- Requirements for protecting audit records stored on the TOE are specified in FAU_STG.1
- Requirements for secure transmission of local audit records to an external IT entity via a secure channel are specified in FAU_STG_EXT.1
- Additional requirements for dealing with potential loss of locally stored audit records are specified in FAU_STG.3/LocSpace
- Configuration of the audit functionality is specified in FMT_SMF.1, and confining this functionality to Security Administrators is required by FMT_MOF.1/Functions.

3.2.7 T.SECURITY_FUNCTIONALITY_COMPROMISE

Threat agents may compromise credentials and device data enabling continued access to the network device and its critical data. The compromise of credentials includes replacing existing credentials with an attacker's credentials, modifying existing credentials, or obtaining the Administrator or device credentials for use by the attacker.

SFR Rationale:

- Protection of secret/private keys against compromise is specified in FPT_SKP_EXT.1
- Secure destruction of keys is specified in FCS_CKM.4
- Management of keys is specified in FMT_SMF.1, and confining this functionality to Security Administrators is required by FMT_MTD.1/CryptoKeys
- (Protection of passwords is separately covered under T.PASSWORD_CRACKING),

3.2.8 T.PASSWORD_CRACKING

Threat agents may be able to take advantage of weak administrative passwords to gain privileged access to the device. Having privileged access to the device provides the attacker unfettered access to the network traffic, and may allow them to take advantage of any trust relationships with other network devices.

SFR Rationale:

- Requirements for password lengths and available characters are set in FIA_PMG_EXT.1
- Protection of password entry by providing only obscured feedback is specified in FIA_UAU.7
- Actions on reaching a threshold number of consecutive password failures are specified in FIA_AFL.1
- Requirements for secure storage of passwords are set in FPT_APW_EXT.1.

3.2.9 T.SECURITY_FUNCTIONALITY_FAILURE

A component of the network device may fail during start-up or during operations causing a compromise or failure in the security functionality of the network device, leaving the device susceptible to attackers.

SFR Rationale:

- Requirements for running self-test(s) are defined in FPT_TST_EXT.1

3.3 Assumptions

This section describes the assumptions made in identification of the threats and security requirements for network devices. The network device is not expected to provide assurance in any of these areas, and as a result, requirements are not included to mitigate the threats associated.

3.3.1 A.PHYSICAL_PROTECTION

The network device is assumed to be physically protected in its operational environment and not subject to physical attacks that compromise the security and/or interfere with the device's physical interconnections and correct operation. This protection is assumed to be sufficient to protect the device and the data it contains. As a result, the ST will not include any requirements on physical tamper protection or other physical attack mitigations. The ST will not expect the product to defend against physical access to the device that allows unauthorized entities to extract data, bypass other controls, or otherwise manipulate the device.

[OE.PHYSICAL]

3.3.2 A.LIMITED_FUNCTIONALITY

The device is assumed to provide networking functionality as its core function and not provide functionality/services that could be deemed as general purpose computing. For example, the device should not provide a computing platform for general purpose applications (unrelated to networking functionality).

[OE.NO_GENERAL_PURPOSE]

3.3.3 A.NO_THRU_TRAFFIC_PROTECTION

A standard/generic network device does not provide any assurance regarding the protection of traffic that traverses it. The intent is for the network device to protect data that originates on or is destined to the device itself, to include administrative data and audit data. Traffic that is traversing the network device, destined for another network entity, is not covered by this ST. It is assumed that this protection will be covered by cPPs for particular types of network devices (e.g., firewall).

[OE.NO_THRU_TRAFFIC_PROTECTION]

3.3.4 A.TRUSTED_ADMINISTRATOR

The Security Administrator(s) for the network device are assumed to be trusted and to act in the best interest of security for the organization. This includes being appropriately trained, following policy, and adhering to guidance documentation. Administrators are trusted to ensure passwords/credentials have sufficient strength and entropy and to lack malicious intent when administering the device. The network device is not expected to be capable of defending against a malicious Administrator that actively works to bypass or compromise the security of the device.

[OE.TRUSTED_ADMIN]

3.3.5 A.REGULAR_UPDATES

The network device firmware and software is assumed to be updated by an Administrator on a regular basis in response to the release of product updates due to known vulnerabilities.

[OE.UPDATES]

3.3.6 A.ADMIN_CREDENTIALS_SECURE

The Administrator’s credentials (private key) used to access the network device are protected by the platform on which they reside.

[OE.ADMIN_CREDENTIALS_SECURE]

3.3.7 A.RESIDUAL_INFORMATION

The Administrator must ensure that there is no unauthorized access possible for sensitive residual information (e.g. cryptographic keys, keying material, PINs, passwords etc.) on networking equipment when the equipment is discarded or removed from its operational environment.

[OE.RESIDUAL_INFORMATION]

3.4 Organizational Security Policies

An organizational security policy is a set of rules, practices, and procedures imposed by an organization to address its security needs.

3.4.1 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.

SFR Rationale:

- An advisory notice and consent warning message is required to be displayed by FTA_TAB.1

[FTA_TAB.1]

4 Security Objectives

4.1 Security Objectives for the Operational Environment

The following subsections describe objectives for the Operational Environment.

4.1.1 OE.PHYSICAL

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

4.1.2 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.

4.1.3 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.

4.1.4 OE.TRUSTED_ADMIN

TOE Administrators are trusted to follow and apply all guidance documentation in a trusted manner.

4.1.5 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.

4.1.6 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.

4.1.7 OE.RESIDUAL_INFORMATION

The Security Administrator ensures that there is no unauthorized access possible for sensitive residual information (e.g. cryptographic keys, keying material, PINs, passwords etc.) on networking equipment when the equipment is discarded or removed from its operational environment.

5 Extended Components Definition

The extended components used in this ST are defined in [CPP_ND]. The following table provide a chapter specific reference in which chapter of [CPP_ND] each of the extended components is defined.

Table 5-1 Definition of Extended Components - references to [CPP_ND]

Extended Component	Defined in [CPP_ND] chap.
Mandatory Requirements (<M>)	
FAU_STG_EXT.1	C.1.1.1
FCS_RBG_EXT.1	C.2.1.1
FIA_PMG_EXT.1	C.3.1.1
FIA_UIA_EXT.1	C.3.2.1
FIA_UAU_EXT.2	C.3.3.1
FPT_SKP_EXT.1	C.4.1.1
FPT_APW_EXT.1	C.4.2.1
FPT_TST_EXT.1	C.4.3.1
FPT_TUD_EXT.1	C.4.4.1
FPT_STM_EXT.1	C.4.5.1
FTA_SSL_EXT.1	C.5.1.1
Optional Requirements (<O>)	
None	None.
Selection-Based Requirements (<S>)	
FCS_SSHC_EXT.1	C.2.2.5
FCS_SSHS_EXT.1	C.2.2.6
FCS_TLSC_EXT.1	B.2.1.5
FIA_X509_EXT.1/Rev	C.3.4.1
FIA_X509_EXT.2	C.3.4.2

6 Security Functional Requirements

Conventions

The conventions used in descriptions of the SFRs are as follows:

- Unaltered SFRs are stated in the form used in [CC2] or their extended component definition (ECD);
- Refinement made in the cPP: the refinement text is indicated with **bold text** and ~~strikethroughs~~;
- Selection wholly or partially completed in the cPP: the selection values (i.e. the selection values adopted in the cPP or the remaining selection values available for the ST) are indicated with underlined text

  e.g. "[selection: *disclosure, modification, loss of use*]" in [CC2] or an ECD might become "disclosure" (completion) or "[selection: disclosure, modification]" (partial completion) in the PP;
- Assignment wholly or partially completed in the cPP: indicated with *italicized text*;
- Assignment completed within a selection in the cPP: the completed assignment text is indicated with *italicized and underlined text*

  e.g. "[selection: *change_default, query, modify, delete, [assignment: other operations]*]" in [CC2] or an ECD might become "change_default, *select_tag*" (completion of both selection and assignment) or "[selection: change_default, *select_tag*, *select_value*]" (partial completion of selection, and completion of assignment) in the PP;
- Iteration: indicated by adding a string starting with "/" (e.g. "FCS_COP.1/Hash"), or by appending the iteration number in parenthesis, e.g. (1), (2), (3).
- Application Notes added by the ST author are called 'Additional Application Note' which are enumerated as 'a', 'b', ... and are formatted with underline such as "Additional Application Note a";
- References: Indicated with [square brackets].

[CPP_ND] distinguishes mandatory requirements from optional requirements and selection-based requirements. This ST will mark mandatory requirements by <M>, optional requirements by <O> and selection-based requirements by <S>.

6.1 Functional Security Requirements

6.1.1 Security Audit (FAU)

6.1.1.1 FAU_GEN.1 Audit data generation<M>

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).*
- *Changes to TSF data related to 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.

d) *Specifically defined auditable events listed in Table 6-1.*

Application Note 1

If the list of 'administrative actions' appears to be incomplete, the assignment in the selection should be used to list additional administrative actions which are audited.

The ST author replaces the cross-reference to the table of audit events with an appropriate cross-reference for the ST. This must also include the relevant parts of Table 3 and Table 4 for optional and selection-based SFRs included in the ST.

The list of 'administrative actions' are audited.

The audit events of optional and selection-based SFRs are included in the following auditable events table.

Application Note 2

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

The TSS should identify what information is logged to identify the relevant key for the administrative task of generating/import of, changing, or deleting of cryptographic keys.

With respect to FAU_GEN.1.1 the term 'services' refers to trusted path and trusted channel communications, on demand self-tests, trusted update and administrator sessions (that exist under the trusted path) (e.g. netconf).

All auditable events directly in the table have its own audit log.

Administrators have the ability to execute CLI command to generate/import of/delete cryptographic keys, each command will generate a log and will be stored in log file.

Starting and stopping the referred service will generate a log for audit.

Additional Application Note a: Audit functionality is enabled by default. The auditing functionality cannot be disabled.

Additional Application Note b: The TOE does not support using reset command to reset password directly, but it can modify password in the following way: re-create local-user or change local-user password.

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 6-1*.

Application Note 3

The ST author replaces the cross-reference to the table of audit events with an appropriate cross-reference for the ST. This must also include the relevant parts of table 3 and Table 4 for optional and selection-based SFRs included in the ST.

Date and time of the event, type of event, subject identity, and the outcome of the event are included in audit log.

The optional and selection-based SFRs are included in the following auditable events Table 6-1.

Table 6-1 Security Functional Requirements and Auditable Events

Requirement	**Auditable Events**	**Additional Audit Record Contents**
Mandatory Requirements (<M>)		
FAU_GEN.1	None.	None.
FAU_GEN.2	None.	None.
FAU_STG_EXT.1	None.	None.
FCS_CKM.1	None.	None.
FCS_CKM.2	None.	None.
FCS_CKM.4	None.	None.
FCS_COP.1/DataEncryption	None.	None.
FCS_COP.1/SigGen	None.	None.
FCS_COP.1/Hash	None.	None.
FCS_COP.1/KeyedHash	None.	None.
FCS_RBG_EXT.1	None.	None.
FIA_AFL.1	Unsuccessful login attempts limit is met or exceeded.	Origin of the attempt (e.g., IP address).
FIA_PMG_EXT.1	None.	None.
FIA_UIA_EXT.1	All use of the identification and authentication mechanism.	Origin of the attempt (e.g., IP address).
FIA_UAU_EXT.2	All use of the identification and authentication mechanism.	Origin of the attempt (e.g. IP address).
FIA_UAU.7	None.	None.
FMT_MOF.1/ManualUpdate	Any attempt to initiate a manual update	None.
FMT_MTD.1/CoreData	All management activities of TSF data.	None.
FMT_SMF.1	None.	None.
FMT_SMR.2	None.	None.
FPT_SKP_EXT.1	None.	None.
FPT_APW_EXT.1	None.	None.
FPT_TST_EXT.1	None.	None
FPT_TUD_EXT.1	Initiation of update; result of the update attempt (success or failure).	None
FPT_STM_EXT.1	Discontinuous changes to time -	For discontinuous changes

Requirement	Auditable Events	Additional Audit Record Contents
	either Administrator actuated or changed via an automated process. (Note that no continuous changes to time need to be logged. See also application note on FPT_STM_EXT.1)	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).
FTA_SSL_EXT.1	The termination of a local session by the session locking mechanism.	None.
FTA_SSL.3	The termination of a remote session by the session locking mechanism.	None.
FTA_SSL.4	The termination of an interactive session.	None
FTA_TAB.1	None.	None.
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/Admin	Initiation of the trusted path. Termination of the trusted path. Failures of the trusted path functions.	None
Optional Requirements (<O>)		
FAU_STG.1	None.	None.
FAU_STG.3/LocSpace	Low storage space for audit events.	None.
FMT_MOF.1/Services	Starting and stopping of services.	None.
FMT_MTD.1/CryptoKeys	Management of cryptographic keys.	None.
Selection-Based Requirements (<S>)		
FCS_SSHC_EXT.1	Failure to establish an SSH session.	Reason for failure.
FCS_SSHS_EXT.1	Failure to establish an SSH session.	Reason for failure.
FCS_TLSC_EXT.1	Failure to establish a TLS Session.	Reason for failure.
FIA_X509_EXT.1/Rev	Unsuccessful attempt to validate a certificate.	Reason for failure.
FIA_X509_EXT.2	None.	None.
FIA_X509_EXT.3	None.	None.
FMT_MOF.1/Functions	Modification of the behaviour of the transmission of audit data to an external IT entity, the handling of audit data, the audit functionality when Local Audit Storage Space is full.	None.

Application Note 4

Additional audit events will apply to the TOE depending on the optional and selection-based requirements adopted from Appendix A and Appendix B. The ST author must therefore include the relevant additional events specified in the tables Table 4 and Table 5.

Application Note 39 has been omitted since it relates only to an optional SFR not contained in this ST.

Application Note 5

The audit event for FIA_X509_EXT.1/Rev is based on the TOE not being able to complete the certificate validation by ensuring the following:

- *the presence of the basicConstraints extension and that the CA flag is set to TRUE for all CA certificates.*
- *Verification of the digital signature of the trusted hierarchical CA*
- *read/access the CRL or access the OCSP server (according to selections in the ST).*

If any of these checks fails, then an audit event with the failure should be written to the audit log.

All the relevant additional events specified in the tables in Appendix A and Appendix B of [CPP_ND] are included in the audit event Table 6-1.

6.1.1.2 FAU_GEN.2 User identity association<M>

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.

6.1.1.3 FAU_STG_EXT.1 Protected Audit Event Storage<M>

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.

Application Note 6

For selecting the option of transmission of generated audit data to an external IT entity the TOE relies on a non-TOE audit server for storage and review of 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 in that case. Since the external audit server is not part of the TOE, there are no requirements on it except the capabilities for FTP_ITC.1 transport for audit data. No requirements are placed upon the format or underlying protocol of the audit data being transferred. The TOE must be capable of being configured to transfer audit data to an external IT entity without Administrator intervention. Manual transfer would not meet the requirements. Transmission could be done in real-time or periodically. If the transmission is not done in real-time then the TSS describes what event stimulates the transmission to be made and what range of frequencies the TOE supports for making transfers of audit data to the audit server; the TSS also suggests typical acceptable frequencies for the transfer.

The TOE supports transmitting the audit data to syslog server via trust channel.

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***:*** **[***overwrite the oldest log information always***]]** when the local storage space for audit data is full.

Application Note 7

The external log server might be used as alternative storage space in case the local storage space is full. The 'other action' could in this case be defined as 'send the new audit data to an external IT entity'.

The TOE shall overwrite oldest audit records when the local storage space for audit data is full.

6.1.1.4 FAU_STG.3/LocSpace Action in case of possible audit data loss < O >

FAU_STG.3.1/LocSpace The TSF shall generate a warning to inform the Administrator if the audit trail exceeds the local audit trail storage capacity.

Additional Application Note c: The local storage that store audit data is CF card.

Application Note 8

This option should be chosen if the TOE generates a warning to inform the Administrator before the local storage space for audit data is used up. This might be useful if auditable events are stored on local storage space only.
It has to be ensured that the warning message required by FAU_STG.3.1/LocSpace can be communicated to the user. The communication should be done via the audit log itself because it cannot be guaranteed that an administrative session is active at the time the event occurs.

The warning should inform the Administrator when the local space to store audit data is used up and/or the TOE will lose audit data due to insufficient local space.

If FAU_STG.3/LocSpace is added to the ST, the ST has to make clear any situations in which audit records might be "invisibly lost".

A warning event will be generated when the local space is insufficient, and the warning event will be sent to NMS.

6.1.1.5 FAU_STG.1 Protected audit trail storage <O>

FAU_STG.1.1 The TSF shall protect the stored audit records in the audit trail from unauthorized deletion.

FAU_STG.1.2 The TSF shall be able to prevent unauthorized modifications to the stored audit records in the audit trail.

An administrator cannot alter audit records but can delete audit information records as a whole.

6.1.2 Cryptographic Support (FCS)

6.1.2.1 FCS_CKM.1 Cryptographic Key Generation (Refinement) <M>

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;**

] ~~and specified cryptographic key sizes [assignment: cryptographic key sizes] that meet the following: [assignment: list of standards].~~

Application Note 9

The ST author selects all key generation schemes used for key establishment and device authentication. When key generation is used for key establishment, the schemes in FCS_CKM.2.1 and selected cryptographic protocols must match the selection. When key generation is used for device authentication, other than ssh-rsa, ecdsa-sha2-nistp256, ecdsa-sha2-nistp384 and ecdsa-sha2-nistp521, the public key is expected to be associated with an X.509v3 certificate.

If the TOE acts as a receiver in the key establishment schemes and is not configured to support mutual authentication, the TOE does not need to implement key generation.

In a distributed TOE, if the TOE component acts as a receiver in the key establishment scheme, the TOE does not need to implement key generation.

The key generation scheme of RSA is used for key establishment by TLS.

6.1.2.2 FCS_CKM.2 Cryptographic Key Establishment (Refinement)<M>

FCS_CKM.2.1 The TSF shall **perform** cryptographic **key establishment** in accordance with a specified cryptographic key **establishment** method: [

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

] ~~that meets the following: [assignment: list of standards].~~

Application Note 10

This is a refinement of the SFR FCS_CKM.2 to deal with key establishment rather than key distribution.

The ST author selects all key establishment schemes used for the selected cryptographic protocols. For Diffie-Hellman group 14, ST authors should make the corresponding selection from the SFR instead of using the Finite field-based key establishment selection.

The RSA-based key establishment schemes are described in Section 9 of NIST SP 800-56B ***Revision 1****; however, Section 9 relies on implementation of other sections in SP 800-56B* ***Revision 1****.*

The elliptic curves used for the key establishment scheme correlate with the curves specified in FCS_CKM.1.1.

The domain parameters used for the finite field-based key establishment scheme are specified by the key generation according to FCS_CKM.1.1.

The DH key establishment is supported by TLS.

6.1.2.3 FCS_CKM.4 Cryptographic Key Destruction<M>

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 [
  - logically addresses the storage location of the key and performs a [single, overwrite consisting of a new value of the key;]]

]

that meets the following: *No Standard.*

Application Note 11

In parts of the selections where keys are identified as being destroyed by "a part of the TSF", the TSS identifies the relevant part and the interface involved. The interface referenced in the requirement could take different forms for different TOEs, the most likely of which is an application programming interface to an OS kernel. There may be various levels of abstraction visible. For instance, in a given implementation the application may have access to the file system details and may be able to logically address specific memory locations. In another implementation the application may simply have a handle to a resource and can only ask another part of the TSF such as the interpreter or OS to delete the resource.

Where different key destruction methods are used for different keys and/or different destruction situations then the different methods and the keys/situations they apply to are described in the TSS (and the ST may use separate iterations of the SFR to aid clarity). The TSS describes all relevant keys used in the implementation of SFRs, including cases where the keys are stored in a non-plaintext form. In the case of non-plaintext storage, the encryption method and relevant key-encrypting-key are identified in the TSS.

Some selections allow assignment of "a value that does not contain any CSP". This means that the TOE uses some specified data not drawn from an RBG meeting FCS_RBG_EXT requirements, and not being any of the particular values listed as other selection options. The point of the phrase "does not contain any CSP" is to ensure that the overwritten data is carefully selected, and not taken from a general pool that might contain current or residual data that itself requires confidentiality protection.

For the avoidance of doubt: the "cryptographic keys" in this SFR include session keys. Key destruction does not apply to the public component of asymmetric key pairs.

SSH or TLS session keys for encryption are in the volatile memory and will be destructed by a single overwrite of zeroes.

SSH private host keys for signature or authentication are saved in internal flash and can be destructed by a single overwrite with a new value of the key by a command.

TLS private keys for signature or authentication are saved in internal flash and can be destructed by a single overwrite with a new value of the key by a command.

Radius shared secrets for authentication are saved in internal flash and can be destructed by a single overwrite with a new value of the key by a command.

6.1.2.4 FCS_COP.1/DataEncryption Cryptographic Operation (AES Data Encryption/ Decryption)<M>

FCS_COP.1.1(1) The TSF shall perform *encryption/decryption* in accordance with a specified cryptographic algorithm *AES used in* [GCM] *mode* and cryptographic key sizes : [128 bits, 256 bits] that meet the following: *AES as specified in ISO 18033-3,* [GCM as specified in ISO 19772].

Application Note 12

For the first selection of FCS_COP.1.1 /DataEncryption, the ST author chooses the mode or modes in which AES operates. For the second selection, the ST author chooses the key sizes that are supported by this functionality. The modes and key sizes selected here correspond to the cipher suite selections made in the trusted channel requirements.

ASE128 and AES256 correspond to the cipher suite selections made in the trusted channel requirements. These trusted channels include TLS, and SSH.

6.1.2.5 FCS_COP.1/SigGen Cryptographic Operation (Signature Generation and Verification)<M>

FCS_COP.1.1/SigGen 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*] ,

]

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 RSASSA-PKCS1v1_5; ISO/IEC 9796-2, Digital signature scheme 2 or Digital Signature scheme 3,

].

Application Note 13

The ST Author chooses the algorithm(s) implemented to perform digital signatures. For the algorithm(s) chosen, the ST author makes the appropriate assignments/selections to specify the parameters that are implemented for that algorithm. The ST author ensures that the assignments and selections for this SFR include all the parameter values necessary for the cipher suites selected for the protocol SFRs (see Appendix B.2.1) that are included in the ST. The ST Author checks for consistency of selections with other FCS requirements, especially when supporting elliptic curves.

The TSF supports RSA Digital Signature Algorithm and cryptographic key sizes 2048 bits. SSH supports RSA Digital Signature Algorithm.

TLS supports RSA Digital Signature Algorithm.

6.1.2.6 FCS_COP.1/Hash Cryptographic Operation (Hash Algorithm) <M>

FCS_COP.1.1(3) The TSF shall perform *cryptographic hashing services* in accordance with a specified cryptographic algorithm: [SHA-1, SHA-256, SHA-384] ~~and cryptographic key sizes [assignment: cryptographic key sizes]~~ and message digest sizes [160, 256, 384] bits that meet the following: *ISO/IEC 10118-3:2004*.

Application Note 14

Vendors are strongly encouraged to implement updated protocols that support the SHA-2 family; until updated protocols are supported, this PP allows support for SHA-1 implementations in compliance with SP 800-131A. In a future version of this cPP, SHA-256 will be the minimum requirement for all TOEs.

The hash selection should be consistent with the overall strength of the algorithm used for FCS_COP.1/DataEncryption and FCS_COP.1/SigGen (for example, SHA 256 for 128-bit keys).

SSH supports cryptographic hashing algorithm of SHA-1 and SHA-256.

TLS supports cryptographic hashing algorithm of SHA-1, SHA-256 and SHA-384.

6.1.2.7 FCS_COP.1/KeyedHash Cryptographic Operation (Keyed Hash Algorithm) <M>

FCS_COP.1.1/KeyedHash The TSF shall perform *keyed-hash message authentication* in accordance with a specified cryptographic algorithm: [HMAC-SHA-1, HMAC-SHA-256] and cryptographic key sizes: [*160 bits for HMAC-SHA-1, 256 bits for HMAC-SHA-256*] **and message digest sizes: [160, 256]** **bits** that meet the following: *ISO/IEC 9797-2:2011, Section 7 "MAC Algorithm 2"*.

Application Note 15

The key size [k] in the assignment falls into a range between L1 and L2 (defined in ISO/IEC 10118 for the appropriate hash function). For example, for SHA-256, L1=512, L2=256, where L2<=k<=L1.

SSH performs keyed-hash message authentication in accordance with the specified cryptographic algorithm: HMAC-SHA-1, HMAC-SHA-256.

TLS performs keyed-hash message authentication in accordance with the specified cryptographic algorithm: HMAC-SHA-1, HMAC-SHA-1-96, HMAC-SHA-256.

6.1.2.8 FCS_RBG_EXT.1 Random Bit Generation<M>

FCS_RBG_EXT.1.1 The TSF shall perform all deterministic random bit generation services in accordance with ISO/IEC 18031:2011 using [Hash_DRBG (any)].

Additional Application Note d: This bit generation services is also in accordance with AIS20 with a deterministic random number generator of class DRG.3 and therefore fulfills the following SFRs:

FCS_RNG.1.1 The TSF shall provide a ***deterministic*** random number generator that implements:

- DRG.3.1: If initialized with a random seed of 32 bytes that shall contain at least 128 bit entropy the internal state of the RNG shall have Min-entropy of at least 100 bits.
- DRG.3.2: The DRNG provides forward secrecy.
- DRG.3.3: The DRNG provides backward secrecy even if the current internal state is known.

FCS_RNG.1.2 The TSF shall provide random numbers that meet:

- DRG.3.4: The RNG, initialized with a random seed of 256 bits during the preparative operations ensured by the preparative procedures for users generates output for which more than 2^{14} strings of bit length 128 are mutually different with probability greater than $1-2^{-8}$.
- DRG.3.5: Statistical test suites cannot practically distinguish the random numbers from output sequences of an ideal RNG. The random numbers must pass test procedure A and no other test suites.

FCS_RBG_EXT.1.2 The deterministic RBG shall be seeded by at least one entropy source that accumulates entropy from [[*1*] hardware-based noise source] with a minimum of [256 bits] of entropy at least equal to the greatest security strength, according to ISO/IEC 18031:2011of the keys and CSPs that it will generate.

Application Note 16

For the first selection in FCS_RBG_EXT.1.2, the ST author selects at least one of the types of noise sources. If the TOE contains multiple noise sources of the same type, the ST author fills the assignment with the appropriate number for each type of source (e.g., 2 software-based noise sources, 1 hardware-based noise source). The documentation and tests required in the Evaluation Activity for this element should be repeated to cover each source indicated in the ST.

ISO/IEC 18031:2011 contains three 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, and include the specific underlying cryptographic primitives used in the requirement. While any of the identified hash functions (SHA-1, SHA-256, SHA-384, SHA-512) are allowed for Hash_DRBG or HMAC_DRBG, only AES-based implementations for CTR_DRBG are allowed.
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, which must be equal or greater than the security strength of any key generated by the TOE.
TSF uses Hash_DRBG to perform deterministic random bit generation. The identified hash functions (SHA-1, SHA-256) are allowed for Hash_DRBG

Additional application note e:
The internal noise source is used to generate random number. The operations have been performed according to chap. 4.8 of [AIS20]. For certified use of the TOE, the random seed of 32 bytes shall contain at least 128 bit entropy. The seed value is provided by the hardware TPM chip integrated in the SCC. The random numbers from the integrated TPM chip are only used to seed the DRBG. The TOE ensures that TLS-based communication and SSH-based communication cannot be enabled before the DRNG is seeded.

The DRNG complies with ISO/IEC 18031:2011 using Hash_DRBG (SHA-256).

6.1.2.9 FCS_SSHC_EXT.1 SSH Client Protocol<S>

FCS_SSHC_EXT.1.1 The TSF shall implement the SSH protocol that complies with RFC(s) [4251, 4252, 4253, 4254, 6668].

Application Note 17

The ST author selects which of the 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). RFC 4253 indicates that certain cryptographic algorithms are "REQUIRED". This means that the implementation must include support, not that the algorithms must be enabled for use. Ensuring that algorithms indicated as "REQUIRED" but not listed in the later elements of this component are implemented is out of scope of the evaluation activity for this requirement.
All required algorithms required by RFC 4253 are supported.

FCS_SSHC_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.

For password-based authentication the guidance documentation will specify a minimum length for the password to ensure a minimum security strength of 100 bit (~16-20 characters, to be checked).

FCS_SSHC_EXT.1.3 The TSF shall ensure that, as described in RFC 4253, packets greater than [*262144*] bytes in an SSH transport connection are dropped.

Application Note 18

RFC 4253 provides for the acceptance of "large packets" with the caveat that the 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.

The TOE drops packets greater than 256 KB in an SSH transport connection. Packets of size greater than 35000 bytes and smaller than 256 KB are not dropped because of that the TOE may support uncompressed big certificates.

FCS_SSHC_EXT.1.4 The TSF shall ensure that the SSH transport implementation uses the following encryption algorithms and rejects all other encryption algorithms: [AEAD_AES_128_GCM, AEAD_AES_256_GCM].

Application Note 19

RFC 5647 specifies the use of the AEAD_AES_128_GCM and AEAD_AES_256_GCM algorithms in SSH. As described in RFC 5647, AEAD_AES_128_GCM and AEAD_AES_256_GCM can only be chosen as encryption algorithms when the same algorithm is being used as the MAC algorithm. Corresponding FCS_COP entries are included in the ST for the algorithms selected here.

The TOE supports encryption in AES-GCM mode.

FCS_SSHC_EXT.1.5 The TSF shall ensure that the SSH public-key based authentication implementation uses [ssh-rsa] and [no other public key algorithms] as its public key algorithm(s) and rejects all other public key algorithms.

Application Note 20

*If x509v3-ecdsa-sha2-nistp256, x509v3-ecdsa-sha2-nistp384 or x509v3-ecdsa-sha2-nistp521 are selected, then the list of trusted certification authorities must be selected in FCS_SSHC_EXT.1.9 and the FIA_X509_EXT SFRs in Appendix B are applicable.*The SSH transport implementation uses ssh-rsa as its public key algorithm. And it doesn’t support x509v3.

FCS_SSHC_EXT.1.6 The TSF shall ensure that the SSH transport implementation uses [hmac-sha1, hmac-sha1-96, hmac-sha2-256] *and* [no other MAC algorithms] as its data integrity MAC algorithm(s) and rejects all other MAC algorithm(s).

Application Note 21

RFC 5647 specifies the use of the AEAD_AES_128_GCM and AEAD_AES_256_GCM algorithms in SSH. As described in RFC 5647, AEAD_AES_128_GCM and AEAD_AES_256_GCM can only be chosen as MAC algorithms when the same algorithm is being used as the encryption algorithm. RFC 6668 specifies the use of the sha2 algorithms in SSH.

The SSH transport implementation uses hmac-sha1, hmac-sha1-96, hmac-sha2-256 as its data integrity MAC algorithm. And it doesn’t support all other MAC algorithm.

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

FCS_SSHC_EXT.1.8 The TSF shall ensure that within SSH connections the same session keys are used for a threshold of no longer than one hour, and no more than one gigabyte of transmitted data. After either of the thresholds are reached a rekey needs to be performed.

Application Note 22

This SFR defines two thresholds - one for the maximum time span the same session keys can be used and the other one for the maximum amount of data that can be transmitted using the same session keys. Both thresholds need to be implemented and a rekey needs to be performed on whichever threshold is reached first. For the maximum transmitted data threshold, the total incoming and outgoing data needs to be counted. The rekey applies to all session keys (encryption, integrity protection) for incoming and outgoing traffic.

It is acceptable for a TOE to implement lower thresholds than the maximum values defined in the SFR.

For any configurable threshold related to this requirement the guidance documentation needs *to specify how the threshold can be configured. The allowed values must either be specified in the guidance documentation and must be lower or equal to the thresholds specified in this SFR or the TOE must not accept values beyond the thresholds specified in this SFR.*

FCS_SSHC_EXT.1.9 The TSF shall ensure that the SSH client authenticates the identity of the SSH server using a local database associating each host name with its corresponding public key or [no other methods] as described in RFC 4251 section 4.1.

Application Note 23

The list of trusted certification authorities can only be selected if x509v3-ecdsa-sha2-nistp256, x509v3-ecdsa-sha2-nistp384 or x509v3-ecdsa-sha2-nistp521 are selected in FCS_SSHC_EXT.1.5.

These two certificates of x509v3-ecdsa-sha2-nistp256 and x509v3-ecdsa-sha2-nistp384 are not selected in FCS_SSHC_EXT.1.5.

6.1.2.10 FCS_SSHS_EXT.1 SSH Server Protocol <S>

FCS_SSHS_EXT.1.1 The TSF shall implement the SSH protocol that complies with RFC(s) [4251, 4252, 4253, 4254, 6668].

Application Note 24

The ST author selects which of the 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). RFC 4253 indicates that certain cryptographic algorithms are "REQUIRED". This means that the implementation must include support, not that the algorithms must be enabled for use. Ensuring that algorithms indicated as "REQUIRED" but not listed in the later elements of this component are implemented is out of scope of the assurance activity for this requirement.

AES128-CBC and AES256-CBC are implemented according to RFC 4253.

FCS_SSHS_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.

FCS_SSHS_EXT.1.3 The TSF shall ensure that, as described in RFC 4253, packets greater than [*262144*] bytes in an SSH transport connection are dropped.

Application Note 25

RFC 4253 provides for the acceptance of "large packets" with the caveat that the 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.

The TOE drops packets greater than 256 KB in an SSH transport connection. Packets of size greater than 35000 bytes and smaller than 256 KB are not dropped because of that the TOE may support uncompressed big certificates.

FCS_SSHS_EXT.1.4 The TSF shall ensure that the SSH transport implementation uses the following encryption algorithms and rejects all other encryption algorithms: [AEAD_AES_128_GCM, AEAD_AES_256_GCM].

Application Note 26

RFC 5647 specifies the use of the AEAD_AES_128_GCM and AEAD_AES_256_GCM algorithms in SSH. As described in RFC 5647, AEAD_AES_128_GCM and AEAD_AES_256_GCM can only be chosen as encryption algorithms when the same algorithm is being used as the MAC algorithm. Corresponding FCS_COP entries are included in the ST for the algorithms selected here.

The TOE supports encryption in AES-GCM mode.

FCS_SSHS_EXT.1.5 The TSF shall ensure that the SSH public-key based authentication implementation uses [ssh-rsa] *and* [no other public key algorithms] as its public key algorithm(s) and rejects all other public key algorithms.

Application Note 27

If x509v3-ecdsa-sha2-nistp256, x509v3-ecdsa-sha2-nistp384 or x509v3-ecdsa-sha2-nistp521 are selected then the FIA_X509_EXT SFRs in Appendix B are applicable.

The SSH transport implementation uses ssh-rsa as its public key algorithm. And it doesn’t support x509v3.

FCS_SSHS_EXT.1.6 The TSF shall ensure that the SSH transport implementation uses [hmac-sha1, hmac-sha1-96, hmac-sha2-256] *and* [no other MAC algorithms] as its MAC algorithm(s) and rejects all other MAC algorithm(s).

Application Note 28

RFC 5647 specifies the use of the AEAD_AES_128_GCM and AEAD_AES_256_GCM algorithms in SSH. As described in RFC 5647, AEAD_AES_128_GCM and AEAD_AES_256_GCM can only be chosen as MAC algorithms when the same algorithm is being used as the encryption algorithm. RFC 6668 specifies the use of the sha2 algorithms in SSH.

The SSH transport implementation uses hmac-sha1, hmac-sha1-96, hmac-sha2-256 as its data integrity MAC algorithm. And it doesn’t support all other MAC algorithm.

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

FCS_SSHS_EXT.1.8 The TSF shall ensure that within SSH connections the same session keys are used for a threshold of no longer than one hour, and no more than one gigabyte of transmitted data. After either of the thresholds are reached a rekey needs to be performed.

Application Note 29

This SFR defines two thresholds - one for the maximum time span the same session keys can be used and the other one for the maximum amount of data that can be transmitted using the same session keys. Both thresholds need to be implemented and a rekey needs to be performed on whichever threshold is reached first. For the maximum transmitted data threshold, the total incoming and outgoing data needs to be counted. The rekey applies to all session keys (encryption, integrity protection) for incoming and outgoing traffic.

It is acceptable for a TOE to implement lower thresholds than the maximum values defined in the SFR.

For any configurable threshold related to this requirement the guidance documentation needs to specify how the threshold can be configured. The allowed values must either be specified in the guidance documentation and must be lower or equal to the thresholds specified in this SFR or the TOE must not accept values beyond the thresholds specified in this SFR.

6.1.2.11 FCS_TLSC_EXT.1 TLS Client Protocol <S>

FCS_TLSC_EXT.1.1 The TSF shall implement [TLS 1.2 (RFC 5246), TLS 1.1 (RFC 4346)] and reject all other TLS and SSL versions. The TLS implementation will supporting the following ciphersuites:

- [
  - TLS_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5288
  - TLS_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288

  ].

Application Note 30

The ciphersuites to be tested in the evaluated configuration are limited by this requirement. The ST author should select the ciphersuites that are supported. It is necessary to limit the ciphersuites that can be used in an evaluated configuration administratively on the server in the test environment. TLS_RSA_WITH_AES_128_CBC_SHA is not mandatory for ND cPP v2.0 compliance; however, it is required if claiming compliance with RFC 5246.

These requirements will be revisited as new TLS versions are standardized by the IETF.

In a future version of this cPP TLS v1.2 will be required for all TOEs.

The TSF implement TLS1.2 (RFC5246) and the ciphersuites specified in RFC5246 are all selected.

FCS_TLSC_EXT.1.2 The TSF shall verify that the presented identifier matches the reference identifier per RFC 6125 section 6.

Application Note 31

The rules for verification of identify are described in Section 6 of RFC 6125. The reference identifier is established by the Administrator (e.g. entering a URL into a web browser or clicking a link), by configuration (e.g. configuring the name of a mail server or authentication server), or by an application (e.g. a parameter of an API) depending on the application service. Based on a singular reference identifier's source domain and application service type (e.g. HTTP, SIP, LDAP), the client establishes all reference identifiers which are acceptable, such as a Common Name for the Subject Name field of the certificate and a (case-insensitive) DNS name, URI name, and Service Name for the Subject Alternative Name field. The client then compares this list of all acceptable reference identifiers to the presented identifiers in the TLS server's certificate.

The preferred method for verification is the Subject Alternative Name using DNS names, URI names, or Service Names. Verification using the Common Name is required for the purposes of backwards compatibility. Additionally, support for use of IP addresses in the Subject Name or Subject Alternative name is discouraged as against best practices but may be implemented. Finally, the client should avoid constructing reference identifiers using wildcards. However, if the presented identifiers include wildcards, the client must follow the best practices regarding matching; these best practices are captured in the evaluation activity.

The reference identifier is established by the user and by an application (a parameter of an API). Based on a singular reference identifier's source domain and application service type (e.g. HTTP, FTP), the client establishes all reference identifiers including a Common Name for the Subject Name field of the certificate and a (case-insensitive) DNS name for the Subject Alternative Name field. The client then compares this list of all acceptable reference identifiers to the presented identifiers in the TLS server's certificate.

FCS_TLSC_EXT.1.3 The TSF shall only establish a trusted channel if the server certificate is valid. If the server certificate is deemed invalid, then the TSF shall [not establish the connection].

Application Note 32

If TLS is selected in FTP_TRP.1/Admin or FTP_ITC then validity is determined by the identifier verification, certificate path, the expiration date, and the revocation status in accordance with RFC 5280. Certificate validity shall be tested in accordance with testing performed for FIA_X509_EXT.1/Rev. If TLS is selected in FPT_ITT, then certificate validity is tested in accordance with testing performed for FIA_X509_EXT.1/ITT

Validity is determined by the identifier verification, certificate path, the expiration date, and the revocation status in accordance with RFC 5280. Certificate validity is tested in accordance with testing performed for FIA_X509_EXT.1/Rev.

FCS_TLSC_EXT.1.4 The TSF shall [not present the Supported Elliptic Curves Extension] in the Client Hello.

Application Note 33

If ciphersuites with elliptic curves were selected in FCS_TLSC_EXT.1.1, a selection of one or more curves is required. If no ciphersuites with elliptic curves were selected in FCS_TLSC_EXT.1.1, then "none" should be selected.

This requirement limits the elliptic curves allowed for authentication and key agreement to the NIST curves from FCS_COP.1/SigGen and FCS_CKM.1 and FCS_CKM.2. This extension is required for clients supporting Elliptic Curve ciphersuites.

The ciphersuites with elliptic curves were not selected in FCS_TLSC_EXT.1.1. The TSF doesn't support this ciphersuite.

6.1.3 Identification and Authentication (FIA)

6.1.3.1 FIA_AFL.1 Authentication Failure Management (Refinement)<M>

FIA_AFL.1.1 The TSF shall detect when **an Administrator configurable positive integer within** [3 to 5] 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 *unlock* is taken by a local Administrator*;* prevent the offending remote Administrator from successfully authenticating until an Administrator defined time period has elapsed.

Application Note 34

This requirement applies to a defined number of successive unsuccessful authentication attempts and 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.

The TSS describes how the TOE ensures that authentication failures by remote Administrators cannot lead to a situation where no Administrator access is available, either permanently or temporarily (e.g. by providing local logon which is not subject to blocking). The Operational Guidance describes, and identifies the importance of, any actions that are required in order to ensure that Administrator access will always be maintained, even if remote administration is made permanently or temporarily unavailable due to blocking of accounts as a result of FIA_AFL.1.

6.1.3.2 FIA_PMG_EXT.1 Password Management<M>

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 configurable to [8] and [128]..*

Application Note 35

The ST author selects the special characters that are supported by the TOE; they may optionally

list additional special characters supported using the assignment. "Administrative passwords" refers to passwords used by administrators at the local console, over protocols that support passwords, such as SSH and HTTPS, or to grant configuration data that supports other SFRs in the Security Target.
The second assignment should be configured with the largest minimum password length the Security Administrator can configure.
The administrative passwords at local console or over protocols support the same set of special characters that listed in FIA_PMG_EXT.1.1.

6.1.3.3 FIA_UIA_EXT.1 User Identification and Authentication <M>

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;
- [no other actions].

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.

Application Note 36

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 (such as SSH, TLS).
For communications with external IT entities (an audit 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.
According to the application note for FMT_SMR.2, for distributed TOEs at least one TOE component has to support the authentication of Security Administrators according to FIA_UIA_EXT.1 and FIA_UAU_EXT.2 but not necessarily all TOE components. In case not all TOE components support this way of authentication for Security Administrators the TSS shall describe how Security Administrators are authenticated and identified.
Only a banner will show to the user or IT entity and no services are available before authentication.

6.1.3.4 FIA_UAU_EXT.2 Password-based Authentication Mechanism <M>

FIA_UAU_EXT.2.1 The TSF shall provide a local password-based authentication mechanism, [no other authentication mechanism] to perform local administrative user authentication.

Application Note 37

The assignment should be used to identify any additional local authentication mechanisms supported. Local authentication mechanisms are defined as those that occur through the local console; remote administrative sessions (and their associated authentication mechanisms) are specified in FTP_TRP.1/Admin.
According to the application note for FMT_SMR.2, for distributed TOEs at least one TOE component has to support the authentication of Security Administrators according to FIA_UIA_EXT.1 and FIA_UAU_EXT.2 but not necessarily all TOE components. In case not all

TOE components support this way of authentication for Security Administrators the TSS shall describe how Security Administrators are authenticated and identified.

No other authentication mechanism is available except password.

6.1.3.5 FIA_UAU.7 Protected Authentication Feedback <M>

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.

Application Note 38

"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.

The passwords don't display on the screen when inputting the password. Any precise information will not be sent back to the user while the authentication process.

6.1.3.6 FIA_X509_EXT.1/Rev X.509 Certificate Validation <S>

FIA_X509_EXT.1.1/Rev The TSF shall validate certificates in accordance with the following rules:

- RFC 5280 certificate validation and certificate path validation supporting a minimum path length of three certificates.
- 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.*
  - *OCSP certificates presented for OCSP responses shall have the OCSP Signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the extendedKeyUsage field.*

FIA_X509_EXT.1.2/Rev 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.

Application Note 39

FIA_X509_EXT.1.1/Rev lists the rules for validating certificates. The ST author selects whether revocation status is verified using OCSP or CRLs. The trusted channel/path protocols may require that certificates are used; this use requires that the extendedKeyUsage rules are verified. If the TOE does not support functionality that uses any of the certificate types listed in the extendedKeyUsage rules in FIA_X509_EXT.1.1/Rev then this is stated in the TSS and the relevant part of the SFR is considered trivially satisfied. However, if the TOE does support functionality that uses certificates of any of these types then the corresponding rule must be satisfied as in the SFR.

The TOE shall be capable of supporting a minimum path length of three certificates. That is, it shall support a hierarchy comprising of at least a self-signed root certificate, a subordinate CA *certificate and a TOE identity certificate.*

The validation is expected to end in a trusted root CA certificate in a root store managed by the platform.

The TSS shall describe when revocation checking is performed. It is expected that revocation checking is performed when a certificate is used in an authentication step and when performing trusted updates (if selected). It is not sufficient to verify the status of a X.509 certificate only when it is loaded onto the device.

It is not necessary to verify the revocation status of X.509 certificates during power-up self-tests (if the option for using X.509 certificates for self-testing is selected).

FIA_X509_EXT.1.2/Rev applies to certificates that are used and processed by the TSF and restricts the certificates that may be added as trusted CA certificates.

The ST author must include FIA_X509_EXT.1/Rev in all instances except when only SSH is selected within FTP_ITC.1 or FPT_ITT.1 and authentication is limited to ssh-rsa, ecdsa-sha2-nistp256, ecdsa-sha2-nistp384, and/or ecdsa-sha2-nistp521 Additionally, FIA_X509_EXT.1/Rev must also be included if either FPT_TUD_EXT or FPT_TST_EXT have selected to use X509 certificates.

Revocation status is verified using CRLs. TLS requires that certificates are used and this use requires that the extendedKeyUsage rules are verified. The validation is expected to end in a trusted root CA certificate in a root store managed by the platform.

The certificate path must end in a trusted root CA certificate otherwise it will be judged invalid.

6.1.3.7 FIA_X509_EXT.2 X.509 Certificate Authentication <S>

FIA_X509_EXT.2.1 The TSF shall use X.509v3 certificates as defined by RFC 5280 to support authentication for **[**TLS] and [no additional uses].

Application Note 40

In FIA_X509_EXT.2.1, the ST author's selection includes IPsec, TLS, or HTTPS if these protocols are included in FTP_ITC.1.1 or FPT_ITT.1. SSH should be included if authentication other than ssh-rsa, ecdsa-sha2-nistp256, ecdsa-sha2-nistp384, and/or ecdsa-sha2-nistp521 is selected in FCS_SSHC_EXT.1.5 or FCS_SSHS_EXT.1.5. The ST author's selection matches the selection of FTP_ITC.1.1. Certificates may optionally be used for trusted updates of system software (FPT_TUD_EXT.2) and for integrity verification (FPT_TST_EXT.2).

FIA_X509_EXT.2 and FTP_ITC.1.1 do the same selection as they all select TLS.

But certificates are not used for integrity verification (FPT_TST_EXT.2).

FIA_X509_EXT.2.2 When the TSF cannot establish a connection to determine the validity of a certificate; the TSF shall **[**not accept the certificate].

Application Note 41

In FIA_X509_EXT.2.1, the ST author's selection includes IPsec, TLS, or HTTPS if these protocols are included in FTP_ITC.1.1 or FPT_ITT.1. SSH should be included if authentication other than ssh-rsa, ecdsa-sha2-nistp256, ecdsa-sha2-nistp384, and/or ecdsa-sha2-nistp521 is selected in FCS_SSHC_EXT.1.5 or FCS_SSHS_EXT.1.5. The ST author's selection matches the selection of FTP_ITC.1.1. Certificates may optionally be used for trusted updates of system software (FPT_TUD_EXT.2) and for integrity verification (FPT_TST_EXT.2).

Often a connection must be established to check the revocation status of a certificate - either to download a CRL or to perform a lookup using OCSP. The selection is used to describe the behavior in the event that such a connection cannot be established (for example, due to a network error). If the TOE has determined the certificate valid according to all other rules in FIA_X509_EXT.1/Rev, the behavior indicated in the selection determines the validity. The TOE must not accept the certificate if it fails any of the other validation rules in FIA_X509_EXT.1/Rev. If the administrator-configured option is selected by the ST Author, the ST Author also selects the corresponding function in FMT_SMF.1.

If the TOE is distributed and FIA_X509_EXT.1/ITT is selected, then certificate revocation checking is optional. This is due to additional authorization actions being performed in the enabling and disabling of the intra-TOE trusted channel as defined in FCO_CPC_EXT.1. In this case, a connection is not required to determine certificate validity and this SFR is trivially satisfied.

The ST author must include FIA_X509_EXT.2 in all instances except when only SSH is selected within FTP_ITC.1 or FPT_ITT.1 and ssh-rsa, ecdsa-sha2-nistp256, ecdsa-sha2-nistp384, and/or ecdsa-sha2-nistp521 authentication is also selected. Additionally, FIA_X509_EXT.2 must also be included if either FPT_TUD_EXT or FPT_TST_EXT have selected X509 certificates.

The TSF does not accept the certificate when the TSF cannot establish a connection.

6.1.4 Security Management (FMT)

6.1.4.1 FMT_MOF.1/ManualUpdate Management of security functions behaviour <M>

FMT_MOF.1.1/ManualUpdate The TSF shall restrict the ability to enable the functions *to perform manual updates to Security Administrators*.

Application Note 42

FMT_MOF.1/ManualUpdate restricts the initiation of manual updates to Security Administrators.

Only administrators have the privilege to perform manual update.

6.1.4.2 FMT_MOF.1/Functions Management of security functions behaviour<S>

FMT_MOF.1.1/Functions The TSF shall restrict the ability to determine the behaviour of, modify the behaviour of the functions *transmission of audit data to an external IT entity to Security Administrators*.

Application Note 43

FMT_MOF.1/Functions should be chosen if one or more of the following scenarios apply:

- *If the transmission protocol for transmission of audit data to an external IT entity as defined in FAU_STG_EXT.1.1 is configurable, "transmission of audit data to an external IT entity" shall be chosen.*
- *If the handling of audit data is configurable, "handling of audit data" shall be chosen. The term "handling of audit data" refers to the different options for selection and assignments in SFRs FAU_STG_EXT.1.2, FAU_STG_EXT.1.3 and FAU_STG_EXT.2/LocSpace.*
- *If the behaviour of the audit functionality is configurable when Local Audit Storage Space is full, "audit functionality when Local Audit Storage Space is full" shall be chosen.*

The first selection for "determine the behaviour of" and "modify the behaviour of" should be done as appropriate. It might be necessary to have different selections for the first selection depending on the second selection (e.g. "handling of audit data" might require "determine the behaviour of" and "modify the behaviour of" for the first selection on the one hand and "TOE Security Functions" might require "modify the behaviour of" only). In that case FMT_MOF.1/Functions should be iterated with increasing number appended (i.e. FMT_MOF.1/Functions1, FMT_MOF.1/Functions2, etc.).

Only administrators have the privilege to choose the trusted channel for external audit server and decide whether transmit the audit data to an external IT entity or not.

Only administrators have the privilege to modify the behaviour of TOE Security Functions (e.g. cryptographic algorithm, audit server).

6.1.4.3 FMT_MOF.1/Services Management of security functions behaviour <O>

FMT_MOF.1.1/Services The TSF shall restrict the ability to enable and disable the functions and *services to Security Administrators*.

Application Note 44

FMT_MOF.1(2)/AdminAct should only be chosen if the Security Administrator has the ability to start and stop services. In this case the option 'starting and stopping services' shall be chosen in the selection in FAU_GEN.1.1. The term "services" is defined as for FAU_GEN.1.1 (see related Application Notes for FAU_GEN.1.1).

Only administrators have the privilege to enable, disable the functions services (e.g. ssh, radius service).

6.1.4.4 FMT_MTD.1/CoreData Management of TSF Data <M>

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

Application Note 45

The word "manage" includes but is not limited to create, initialize, view, change default, modify, delete, clear, and append. This SFR includes also the resetting of user passwords by the Security Administrator. The identifier "CoreData" has been added here to separate this iteration of FMT_MTD.1 from the optional iteration of FMT_MTD.1 defined in Appendix A.4.2.1 (FMT_MTD.1/CryptoKeys).

Only administrators have the privilege to manage the TSF data (e.g. resetting of user passwords, key generating/deleting, etc).

6.1.4.5 FMT_MTD.1/CryptoKeys Management of TSF data <O>

FMT_MTD.1.1/CryptoKeys The TSF shall restrict the ability to manage the cryptographic keys to Security Administrators.

Application Note 46

FMT_MTD.1.1/CryptoKeys restricts management of cryptographic keys to Security Administrators. It should only be chosen if cryptographic keys can be managed (e.g. modified, deleted or generated/imported) by the Security Administrator. The identifier "CryptoKeys" has been added here to separate this iteration of FMT_MTD.1 from the mandatory iteration of FMT_MTD.1 defined in Chapter 6.6.2.1 (FMT_MTD.1/CoreData).

The administrator can import key or certificate into the TOE and can also delete these keys or certificates.

6.1.4.6 FMT_SMF.1 Specification of Management Functions <M>

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;*

- [
  - Ability to configure audit behavior;
  - Ability to configure thresholds for SSH rekeying.]

Application Note 47

The TOE must provide functionality for both local and remote administration in general. This cPP does not mandate, though, a specific security management function to be available either through the local administration interface, the remote administration interface or both. The TSS shall detail which security management functions are available through which interface(s). The TOE must provide functionality to configure the access banner for FTA_TAB.1 and the session inactivity time(s) for FTA_SSL_EXT.1 and FTA_SSL.3.. The item "Ability to update the TOE, and to verify the updates using digital signature capability prior to installing those updates" includes the relevant management functions from FMT_MOF.1/ManualUpdate, FMT_MOF.1/AutoUpdate (if included in the ST), FIA_X509_EXT.2.2 and FPT_TUD_EXT.1.2 and FPT_TUD_EXT.2.2 (if included in the ST and if they include an administrator-configurable action). Similarly, the selection "Ability to configure audit behavior" includes the relevant management functions from FMT_MOF.1/Services and FMT_MOF.1/Functions, (for all of these SFRs that are included in the ST). If the TOE offers the ability for a remote Administrator account to be disabled in line with FIA_AFL.1 them the ST author should select "Ability to re-enable an Administrator account" to allow the account to be re-enabled by a local Administrator.. If the TOE offers the ability for the administrator to configure the audit behaviour, configure the services available prior to identification or authentication, or if any of the cryptographic functionality on the TOE can be configured, or if the ST is describing a distributed TOE, then the ST author makes the appropriate choice or choices in the second selection, otherwise select "No other capabilities." (in the latter case the selection may alternatively be left blank in the ST).

The selection "Ability to configure thresholds for SSH rekeying" shall be included in the ST if the TOE supports configuration of the thresholds for the mechanisms used to fulfil FCS_SSHC_EXT.1.8 or FCS_SSHS_EXT.1.8 (such configuration then requires the inclusion of FMT_MOF.1/Functions in the ST). If the TOE places limits on the values accepted for the thresholds, then this is stated in the TSS.

The selection "Ability to configure lifetime for IPsec SAs" shall be included in the ST if the TOE supports secure communication via IPsec and the FCS_IPSEC_EXT.1 requirements are included in the ST. The configuration of the lifetime for IPsec SAs needs to be in line with the selection in FCS_IPSEC_EXT.1.7 (such configuration then requires the inclusion of FMT_MOF.1/Functions in the ST).

The selection "Ability to set the time which is used for time-stamps" shall be included in the ST if the TOE allows the Administrator to set the time of the device which is then used in time stamps. This option shall not be selected if the TOE does not allow manual time setting but only relies on synchronization with external time sources like NTP servers.

The selection "Ability to configure the reference identifier for the peer" shall be included in the ST if the TOE supports secure communications via the IPsec protocol and the FCS_IPSEC_EXT.1 requirements are included in the ST. For TOEs that support only IP address and FQDN identifier types, configuration of the reference identifier may be the same as configuration of the peer's name for the purposes of connection.

For distributed TOEs the interaction between TOE components will be configurable (see FCO_CPC_EXT.1). Therefore, the ST author includes the selection "Ability to configure the interaction between TOE components" for distributed TOEs. A simple example would be the change of communication protocol according to FPT_ITT.1. Another example would be changing the management of a TOE component from direct remote administration to remote administration through another TOE component. A more complex use case would be if the realization of an SFR is achieved through two or more TOE components and the responsibilities between the two or more components could be modified.

For distributed TOEs that implement a registration channel (as described in FCO_CPC_EXT.1.2), the ST author uses the selection "Ability to configure the cryptographic functionality" in this SFR, and its corresponding mapping in the TSS, to describe the configuration of any cryptographic aspects of the registration channel that can be modified by the operational environment in order to improve the channel security (cf. the description of the content of Preparative Procedures in [SD, 3.6.1.2]).

The ability to configure the available services before identification and authentication is not supported.

6.1.4.7 FMT_SMR.2 Restrictions on security roles <M>

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.

Application Note 48

FMT_SMR.2.3 requires that a Security Administrator be able to administer the TOE through the local console and through a remote mechanism. The ST Author must select FTP_ITC.1, FPT_ITT.1 and/or FTP_TRP.1/Admin to demonstrate how secure communication is achieved.

For distributed TOEs not every TOE component is required to implement its own user management to fulfil this SFR. At least one component has to support authentication and identification of Security Administrators according to FIA_UIA_EXT.1 and FIA_UAU_EXT.2. For the other TOE components authentication as Security Administrator can be realized through the use of a trusted channel (either according to FTP_ITC.1 or FPT_ITT.1) from a component that supports the authentication of Security Administrators according to FIA_UIA_EXT.1 and FIA_UAU_EXT.2. The identification of users according to FIA_UIA_EXT.1.2 and the association of users with roles according to FMT_SMR.2.2 is done through the components that support the authentication of Security Administrators according to FIA_UIA_EXT.1 and FIA_UAU_EXT.2. TOE components that authenticate Security Administrators through the use of a trusted channel are not required to support local administration of the component as defined in FMT_SMR.2.3.

An administrator is able to admin the TOE through the local console or through a remote mechanism (SSH).

6.1.5 Protection of the TSF (FPT)

6.1.5.1 FPT_SKP_EXT.1 Protection of TSF Data (for reading of all pre-shared, symmetric and private keys) <M>

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

Application Note 49

The intent of this requirement is for the device to protect keys, key material, and authentication credentials from unauthorized disclosure. This data should only be accessed for the purposes *of their assigned security functionality, and there is no need for them to be displayed/accessed at any other time. This requirement does not prevent the device from providing indication that these exist, are in use, or are still valid. It does, however, restrict the reading of the values outright.*

All of the root keys and private keys are stored in flashand only the designed security functionality can access them.

6.1.5.2 FPT_APW_EXT.1 Protection of Administrator Passwords<M>

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.

Application Note 50

The intent of the requirement is that raw password authentication data is 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 could directly read memory to capture a password but is trusted not to do so. . Passwords should be obscured during entry on the local console in accordance with FIA_UAU.7.

The administrator passwords are hashed with salt by SHA256 and won't be stored or displayed in plaintext.

6.1.5.3 FPT_TST_EXT.1 TSF Testing (Extended) <M>

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: [*integrity of the firmware and software (software CRC check), the correct operation of cryptographic functions*].

Application Note 51

It is expected that self-tests are carried out during initial start-up (on power on). Other options should only be used if the developer can justify why they are not carried out during initial start-up. It is expected that at least self-tests for verification of the integrity of the firmware and software as well as for the correct operation of cryptographic functions necessary to fulfil the SFRs will be performed. If not all self-tests are performed during start-up multiple iterations of this SFR are used with the appropriate options selected. In future versions of this cPP the suite of self-tests will be required to contain at least mechanisms for measured boot including self-tests of the components which perform the measurement.

Non-distributed TOEs may internally consist of several components that contribute to enforcing SFRs. Self-testing shall cover all components that contribute to enforcing SFRs and verification of integrity shall cover all software that contributes to enforcing SFRs on all components.

For distributed TOEs all TOE components have to perform self-tests. This does not necessarily mean that each TOE component has to carry out the same self-tests: the ST describes the applicability of the selection (i.e. when self-tests are run) and the final assignment (i.e. which self-tests are carried out) to each TOE component.

Self-test during initial start-up will be performed by the TSF. During this self-test, verification of the integrity of the firmware and software as well as for the correct operation of cryptographic functions will be performed.

Application Note 52

If certificates are used by the self-test mechanism (e.g. for verification of signatures for integrity verification), certificates are validated in accordance with FIA_X509_EXT.1/Rev and should be selected in FIA_X509_EXT.2.1. Additionally, FPT_TST_EXT.2 must be included in the ST.

Certificates are not used by the self-test for verification of signatures for integrity verification.

6.1.5.4 FPT_TUD_EXT.1 Trusted Update <M>

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 [the most recently installed version of the TOE firmware/software].

Application Note 53

If a trusted update can be installed on the TOE with a delayed activation the version of both the currently executing image and the installed but inactive image must be provided. In this case the option 'the most recently installed version of the TOE firmware/software' needs to be chosen from the selection in FPT_TUD_EXT.1.1 and the TSS needs to describe how and when the inactive version becomes active. If all trusted updates become active as part of the installation process, only the currently executing version needs to be provided. In this case the option 'no other TOE firmware/software version' shall be chosen from the selection in FPT_TUD_EXT.1.1..

For a distributed TOE, the method of determining the installed versions on each component of the TOE is described in the operational guidance.

The administrators can query the currently executing version of the TOE firmware/software as well as the most recently installed version by a command. The currently executing patches and most recently installed patches can also be check out.

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].

Application Note 54

The selection in FPT_TUD_EXT.1.2 distinguishes the support of automatic checking for updates and support of automatic updates. The first option refers to a TOE that checks whether a new update is available, communicates this to the Administrator (e.g. through a message during an Administrative session, through log files) but requires some action by the Administrator to actually perform the update. The second option refers to a TOE that checks for updates and automatically installs them upon availability.

The TSS explains what actions are involved in the TOE support when using the “support automatic checking for updates” or “support automatic updates” selections.

When published hash values (see FPT_TUD_EXT.1.3) are used to protect the trusted update mechanism, the TOE must not automatically download the update file(s) together with the hash value (either integrated in the update file(s) or separately) and automatically install the update without any active authorization by the Security Administrator, even when the calculated hash value matches the published hash value. When using published hash values to protect the trusted update mechanism, the option “support of automatic updates” must not be used (automated checking for updates is permitted, though). The TOE may automatically download the update file(s) themselves but not to the hash value. For the published hash approach, it is intended that a Security Administrator is always required to give active authorisation for installation of an update (as described in more detail under FPT_TUD_EXT.1.3) below. Due to this, the type of update mechanism is regarded as “manually initiated update”, even if the update file(s) may be downloaded automatically. A fully automated approach (without Security Administrator intervention) can only be used when ”digital signature mechanism” is selected in FPT_TUD_EXT.1.3 below.

Just manual update is supported

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.

Application Note 55

The digital signature mechanism referenced in the selection of FPT_TUD_EXT.1.3 is one of the algorithms specified in FCS_COP.1/SigGen. The published hash referenced in

FPT_TUD_EXT.1.3 is generated by one of the functions specified in FCS_COP.1/Hash. The ST author should choose the mechanism implemented by the TOE; it is acceptable to implement both mechanisms.

When published hash values are used to secure the trusted update mechanism, an active authorization of the update process by the Security Administrator is always required. The secure transmission of an authentic hash value from the developer to the Security Administrator is one of the key factors to protect the trusted update mechanism when using published hashes and the guidance documentation needs to describe how this transfer has to be performed. For the verification of the trusted hash value by the Security Administrator different use cases are possible. The Security Administrator could obtain the published hash value as well as the update file(s) and perform the verification outside the TOE while the hashing of the update file(s) could be done by the TOE or by other means. Authentication as Security Administrator and initiation of the trusted update would in this case be regarded as "active authorization" of the trusted update. Alternatively, the Administrator could provide the TOE with the published hash value together with the update file(s) and the hashing and hash comparison is performed by the TOE. In case of successful hash verification, the TOE can perform the update without any additional step by the Security Administrator. Authentication as Security Administrator and sending the hash value to the TOE is regarded as "active authorization" of the trusted update (in case of successful hash verification), because the Administrator is expected to load the hash value only to the TOE when intending to perform the update. As long as the transfer of the hash value to the TOE is performed by the Security Administrator, loading of the update file(s) can be performed by the Security Administrator or can be automatically downloaded by the TOE from a repository.

If the digital signature mechanism is selected, the verification of the signature shall be performed by the TOE itself. For the published hash option, the verification can be done by the TOE itself as well as by the Security Administrator. In the latter case use of TOE functionality for the verification is not mandated, so verification could be done using non-TOE functionality of the device containing the TOE or without using the device containing the TOE.

For distributed TOEs all TOE components shall support Trusted Update. The verification of the signature or hash on the update shall either be done by each TOE component itself (signature verification) or for each TOE component (hash verification).

Updating a distributed TOE might lead to the situation where different TOE components are running different software versions. Depending on the differences between the different software versions the impact of a mixture of different software versions might be no problem at all or critical to the proper functioning of the TOE. The TSS shall detail the mechanisms that support the continuous proper functioning of the TOE during trusted update of distributed TOEs.

RSADSA as specified in FCS_COP.1(2) can be used for firmware/software digital signature mechanism to authenticate it prior to installation.

Application Note 56

Future versions of this cPP will mandate the use of a digital signature mechanism for trusted updates.

Digital signature mechanism has been supported by the TSF.

Application Note 57

If certificates are used by the update verification mechanism, certificates are validated in accordance with FIA_X509_EXT.1/Rev and should be selected in FIA_X509_EXT.2.1. Additionally, FPT_TUD_EXT.2 must be included in the ST.

Certificates are not used by the update verification mechanism

Application Note 58

"Update" in the context of this SFR refers to the process of replacing a non-volatile, system resident software component with another. The former is referred to as the NV image, and the latter is the update image. While the update image is typically newer than the NV image, this is not a requirement. There are legitimate cases where the system owner may want to rollback a component to an older version (e.g. when the component manufacturer releases a faulty update,

or when the system relies on an undocumented feature no longer present in the update). Likewise, the owner may want to update with the same version as the NV image to recover from faulty storage.

All discrete software components (e.g. applications, drivers, kernel, firmware) of the TSF, should be digitally signed by the corresponding manufacturer and subsequently verified by the mechanism performing the update. Since it is recognized that components may be signed by different manufacturers, it is essential that the update process verify that both the update and NV images were produced by the same manufacturer (e.g. by comparing public keys) or signed by legitimate signing keys (e.g. successful verification of certificates when using X.509 certificates).

The validation of the firmware/software integrity is always performed before the process of replacing a non-volatile, system resident software component with another is started. All discrete software components (e.g. applications, drivers, kernel, and firmware) of the TSF are archived together into a whole package and the single package is digitally signed.

6.1.5.5 FPT_STM.EXT.1 Reliable Time Stamps <M>

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

FPT_STM_EXT.1.2 The TSF shall allow the Security Administrator to set the time.

Application Note 59

Reliable time stamps are expected to be used with other TSF, e.g. for the generation of audit data to allow the Security Administrator to investigate incidents by checking the order of events and to determine the actual local time when events occurred. The decision about the required level of accuracy of that information is up to the Administrator. The TOE depends on external time and date information, either provided manually by the Security Administrator or through the use of one or more external time sources like NTP servers. The corresponding option(s) shall be chosen from the selection in FPT_STM_EXT.1.2. The use of a local real-time clock and the automatic synchronisation with an external time source (e.g. NTP server) is recommended but not mandated. Note that for the communication with an external time source like an NTP server, the use of FTP_ITC.1 is optional but not mandated. The ST author describes in the TSS how the external time and date information is received by the TOE and how this information is maintained.

The term "reliable time stamps" refers to the strict use of the time and date information, that is provided externally, and the logging of all discontinuous changes to the time settings including information about the old and new time. With this information the real time for all audit data can be determined. Note, that all discontinuous time changes, Administrator actuated or changed via an automated process, must be audited. No audit is needed when time is changed via use of kernel or system facilities – such as daytime (3) – that exhibit no discontinuities in time.

For distributed TOEs it is expected that the Security Administrator ensures synchronization between the time settings of different TOE components. All TOE components shall either be in sync (e.g. through synchronisation between TOE components or through synchronisation of different TOE components with external NTP servers) or the offset should be known to the Administrator for every pair of TOE components. This includes TOE components synchronized to different time zones.

Only administrators have the privilege to configure or modify the time. And the logging of all changes to the time settings includes information about the old and new time. With this information the real time for all audit data can be calculated.

6.1.6 TOE Access (FTA)

6.1.6.1 FTA_SSL_EXT.1 TSF-initiated Session Locking <M>

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

- terminate the session]

after a Security Administrator-specified time period of inactivity.

6.1.6.2 FTA_SSL.3 TSF-initiated Termination (Refinement) <M>

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

6.1.6.3 FTA_SSL.4 User-initiated Termination (Refinement)<M>

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

6.1.6.4 FTA_TAB.1 Default TOE Access Banners (Refinement) <M>

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.

Application Note 60

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.

The administrators can establish a session through SSH and a banner will displayed.

6.1.7 Trusted path/channels (FTP)

6.1.7.1 FTP_ITC.1 Inter-TSF trusted channel (Refined)<M>

FTP_ITC.1.1 The TSF shall be **capable of using [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.

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 service*].

Application Note 61

The intent of the above requirement is to provide a means by which a cryptographic protocol may be used to protect external communications with authorized IT entities that the TOE interacts with to perform its functions. The TOE uses at least one of the listed protocols 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 capable of being protected by one of the listed protocols. If other authorized IT entities are protected, the ST author makes the appropriate assignments (for those entities) and selections (for the protocols that are used to protect those connections). The ST author selects the mechanism or mechanisms supported by the TOE, and then ensures that the detailed protocol requirements in Appendix B corresponding to their selection are included in the ST.*

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 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.

Where public key certificates are used in support of an FTP_ITC.1 channel, FIA_X509_EXT.1/Rev is to be used (this requires checking certificate revocation), and not the iteration FIA_X509_EXT.1/ITT which is only for use in inter-component channels of a distributed TOE.

The TOE establish trust channel with audit server by TLS.

FCS_TLSC_EXT.1 was included in this ST.

6.1.7.2 FTP_TRP.1/Admin Trusted Path (Refinement) <M>

FTP_TRP.1.1/Admin The TSF shall be **capable of using [SSH] 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/Admin The TSF shall permit **remote Administrators** to initiate communication via the trusted path.

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

Application Note 62

This requirement ensures that authorized remote Administrators initiate all communication with the TOE via a trusted path, and that all communication with the TOE by remote administrators is performed over this path. The data passed in this trusted communication channel is encrypted as defined by the protocol chosen in the first selection. The ST author selects the mechanism or mechanisms supported by the TOE, and then ensures that the detailed protocol requirements in Appendix B corresponding to their selection are included in the ST.

The administrators can establish a session through SSH and a banner will display.

6.2 Assurance Security Requirements

The development and the evaluation of the TOE shall be done in accordance to the following security assurance requirements:

Table 6-2 Security Assurance Requirements

Assurance Class	Assurance Components
Security Target (ASE)	Conformance claims (ASE_CCL.1)
	Extended components definition (ASE_ECD.1)
	ST introduction (ASE_INT.1)
	Security objectives for the operational environment (ASE_OBJ.1)
	Stated security requirements (ASE_REQ.1)
	Security Problem Definition (ASE_SPD.1)
	TOE summary specification (ASE_TSS.1)
	TOE summary specification (ASE_TSS.1.1C Refinement)
Development (ADV)	Basic functional specification (ADV_FSP.1)
Guidance documents (AGD)	Operational user guidance (AGD_OPE.1)
	Preparative procedures (AGD_PRE.1)
Life cycle support (ALC)	Labeling of the TOE (ALC_CMC.1)
	TOE CM coverage (ALC_CMS.1)
Tests (ATE)	Independent testing – sample (ATE_IND.1)
Vulnerability assessment (AVA)	Vulnerability survey (AVA_VAN.1)

This security target claims conformance with [CPP_ND]. In addition to [CEM], the evaluation activities for [CPP_ND] are completed in [SD_ND].

6.3 SFR Rationale

The following table lists all SFRs contained in [CPP_ND] together with the classification whether they are mandatory, optional or selection-based, indicates which are included in this ST and provides a dependency rationale. Justifications for any unsupported dependencies will be given in the table as well.

Table 6-3 Dependency rationale for SFRs

Requirement from [CPP_ND]	Dependencies	Satisfied by
Mandatory Requirements (<M>)		

Requirement from [CPP_ND]	Dependencies	Satisfied by
FAU_GEN.1	FPT_STM_EXT.1	FPT_STM_EXT.1 included (which is hierarchic to FPT_STM_EXT.1)
FAU_GEN.2	FAU_GEN.1; FIA_UID.1	FAU_GEN.1; Satisfied by FIA_UIA_EXT.1, which specifies the relevant Administrator identification timing
FAU_STG_EXT.1	FAU_GEN.1; FTP_ITC.1	FAU_GEN.1; FTP_ITC.1
FCS_CKM.1	FCS_CKM.2 or FCS_COP.1; FCS_CKM.4	FCS_CKM.2; FCS_CKM.4
FCS_CKM.2	FTP_ITC.1 or FTP_ITC.2 or FCS_CKM.1; FCS_CKM.4	FCS_CKM.1; FCS_CKM.4
FCS_CKM.4	FTP_ITC.1 or FTP_ITC.2 or FCS_CKM.1	FCS_CKM.1
FCS_COP.1/DataEncryption	FTP_ITC.1 or FTP_ITC.2 or FCS_CKM.1; FCS_CKM.4	FCS_CKM.1; FCS_CKM.4
FCS_COP.1/SigGen	FTP_ITC.1 or FTP_ITC.2 or FCS_CKM.1; FCS_CKM.4	FCS_CKM.1; FCS_CKM.4
FCS_COP.1/Hash	FTP_ITC.1 or FTP_ITC.2 or FCS_CKM.1; FCS_CKM.4	Unsupported Dependencies: This SFR specifies keyless hashing operations, so initialisation and destruction of keys are not relevant
FCS_COP.1/KeyedHash	FTP_ITC.1 or FTP_ITC.2 or FCS_CKM.1; FCS_CKM.4	FCS_CKM.1; FCS_CKM.4
FCS_RBG_EXT.1	None	N/A
FIA_AFL.1	FIA_UAU.1	Satisfied by FIA_UIA_EXT.1, which specifies the relevant Administrator authentication
FIA_PMG_EXT.1	None	N/A
FIA_UIA_EXT.1	FTA_TAB.1	FTA_TAB.1
FIA_UAU_EXT.2	None	N/A
FIA_UAU.7	FIA_UAU.1	Satisfied by FIA_UIA_EXT.1, which specifies the relevant Administrator authentication
FMT_MOF.1/ManualUpdate	FMT_SMR.1; FMT_SMF.1	FMT_SMR.2; FMT_SMF.1
FMT_MTD.1/CoreData	FMT_SMR.1; FMT_SMF.1	FMT_SMR.2; FMT_SMF.1
FMT_SMF.1	None	N/A
FMT_SMR.2	FIA_UID.1	Satisfied by FIA_UIA_EXT.1, which specifies the relevant

Requirement from [CPP_ND]	**Dependencies**	**Satisfied by**
		Administrator identification
FPT_SKP_EXT.1	None	N/A
FPT_APW_EXT.1	None	N/A
FPT_TST_EXT.1	None	N/A
FPT_TUD_EXT.1	FCS_COP.1/SigGen or FCS_COP.1/Hash	FCS_COP.1/SigGen
FPT_STM_EXT.1	None	N/A
FTA_SSL_EXT.1	FIA_UAU.1	Satisfied by FIA_UIA_EXT.1, which specifies the relevant Administrator identification
FTA_SSL.3	None	N/A
FTA_SSL.4	None	N/A
FTA_TAB.1	None	N/A
FTP_ITC.1	None	N/A
FTP_TRP.1/Admin	None	N/A
Optional Requirements (<O>)		
FAU_STG.1	FAU_GEN.1	FAU_GEN.1
FAU_STG.3/LocSpace	FAU_STG.1	FAU_STG.1
FMT_MOF.1/Services	FMT_SMR.1; FMT_SMF.1	FMT_SMR.2; FMT_SMF.1
FMT_MTD.1/CryptoKeys	FMT_SMR.1; FMT_SMF.1	FMT_SMR.2; FMT_SMF.1
Selection-Based Requirements (<S>)		

FCS_SSHC_EXT.1	FCS_CKM.1;
	FCS_CKM.2;
	FCS_COP.1/DataEncryption;
	FCS_COP.1/SigGen;
	FCS_COP.1/Hash;
	FCS_COP.1/KeyedHash;
	FCS_RBG_EXT.1:
		FCS_CKM.1;
		FCS_CKM.2;
		FCS_COP.1/DataEncryption;
		FCS_COP.1/SigGen;
		FCS_COP.1/Hash;
		FCS_COP.1/KeyedHash;
		FCS_RBG_EXT.1:

FCS_SSHS_EXT.1	FCS_CKM.1;
	FCS_CKM.2;
	FCS_COP.1/DataEncryption;
	FCS_COP.1/SigGen;
	FCS_COP.1/Hash;
	FCS_COP.1/KeyedHash;
	FCS_RBG_EXT.1:
		FCS_CKM.1;
		FCS_CKM.2;
		FCS_COP.1/DataEncryption;
		FCS_COP.1/SigGen;
		FCS_COP.1/Hash;
		FCS_COP.1/KeyedHash;
		FCS_RBG_EXT.1:

FCS_TLSC_EXT.1	FCS_CKM.1;
	FCS_CKM.2;
	FCS_COP.1/DataEncryption;
	FCS_COP.1/SigGen;
	FCS_COP.1/Hash;
	FCS_COP.1/KeyedHash;
	FCS_RBG_EXT.1:
		FCS_CKM.1;
		FCS_CKM.2;
		FCS_COP.1/DataEncryption;
		FCS_COP.1/SigGen;
		FCS_COP.1/Hash;
		FCS_COP.1/KeyedHash;
		FCS_RBG_EXT.1:

FIA_X509_EXT.1/Rev	FIA_X509_EXT.2;
	FIA_X509_EXT.3;
		FIA_X509_EXT.2;
		
		According to Network Device Interpretation # 201726, the use of FIA_X509_EXT.3 is optional.

Requirement from [CPP_ND]	Dependencies	Satisfied by

FIA_X509_EXT.2	FIA_X509_EXT.1/Rev;
	FIA_X509_EXT.3;
		FIA_X509_EXT.1/Rev;
		According to Network Device Interpretation # 201726rev2, the use of FIA_X509_EXT.3 is optional.

FMT_MOF.1/Functions	FMT_SMR.1; FMT_SMF.1	FMT_SMR.2; FMT_SMF.1

7 TOE Summary Specification

7.1 Security Audit (FAU)

7.1.1 FAU_GEN.1 Audit data generation

The TOE generates an audit record whenever an audited event occurs. The types of events that cause audit records to be generated include identification and authentication related events, and administrative events (the specific events and the contents of each audit record are listed in the table within the FAU_GEN.1 SFR, "Table 6-1 Auditable Events"). Each of the events specified in the audit record is in enough detail to identify the user for which the event is associated (e.g. user identity, MAC address, IP address), when the event occurred, where the event occurred, the outcome of the event, and the type of event that occurred.

The audit trail consists of the individual audit records; one audit record for each event that occurred. The audit record contains a lot of information, such as the type of event that occurred, and two percent sign (%%), which follows the device name. As noted above, the information includes at least all of the required information. Additional information can be configured and included if desired.

Administrators have the ability to execute CLI command to generate/import of/delete cryptographic keys, each command will generate a log and will be stored in log file.

7.1.2 FAU_GEN.2 User identity association

Each auditable event is associated with the user that triggered the event and as a result, they are traceable to a specific user. For example, a human user, user identity or related session ID would be included in the audit record. For an IT entity or device, the IP address, MAC address, host name, or other configured identification is presented.

The security log of user account management should include user name. Other types of security log have other rules about the information.

7.1.3 FAU_STG.1 Protected audit trail storage

Only the authorized administrators can monitor the logfile record, and operate the log files. The unauthorized users have no access to do those actions. And the actions of the authorized administrators will be logged.

7.1.4 FAU_STG_EXT.1 Protected audit event storage

The TOE supports to export syslog records to a specified, external syslog server. The TOE protects communications with an external syslog server via TLS. The TOE stores audit records on CF card whenever it is connected with syslog server or not. The transmission of audit information to an external syslog server can be done in real-time.

The size of an information file is configurable by the administrator with value 4M/8M/16M/32M bytes. The default maximum size of each information file is 8 MB. When the size of an information file exceeds the configured maximum size, the information file is compressed into a smaller file in standard log_slot ID_time.log.zip format. The maximum quantity of compressed files is configurable by the administrator with a value ranging from 3 to 500. A maximum of 200 files can be stored on a device by default. The unauthorized users are disallowed to handle the audit records.

The logs are saved to flash memory (internal CF card) so records can’t lost in case of failures or restarts. The log buffer is circular, so newer messages overwrite older messages after the buffer is full. Administrators are instructed to monitor the log buffer using the show logging privileged CLI command to view the audit records. The first message displayed is the oldest message in the buffer. There are other associated commands to clear the buffer, to reset log buffer, etc. The size of the log buffer can be configured by users with sufficient privileges.

When the local audit data store in CF card exceeds the maximum allowed size of log file storage, the system deletes oldest compressed files to save the latest log file.

An administrator cannot alter audit records but can delete audit records as a whole.

7.1.5 FAU_STG.3/LocSpace Action in case of possible audit data loss

If the log files have already occupied more than 80% of the total audit storage in CF card, or delete the old log files after saving them to the other storage device, an event will be generated and sent to management server to notice the clients of the warning information.

If the number of compressed log files generated in the system exceeded 80% of the maximum number of compressed files, an event will also be generated to notice net-manager the warning information.

If the number of recorded compressed files reach the maximum number that the security administrator has configured, or the storage with audit events reach the configured storage size, another event will be generated to notice net-manager.

7.2 Cryptographic Support (FCS)

7.2.1 FCS_CKM.1 Cryptographic Key Generation

The TOE’s DRBG is used to generate RSA keys with key sizes 2048 bits according to [FIPS186-4], Appendix B.3.

The generated RSA keys are used for both, key establishment and device authentication.

7.2.2 FCS_CKM.2 Cryptographic Key Establishment

The TOE supports Diffie-Hellman group 14 key establishment. The Hash DRBG is used for every random bits generation from the TOE in the key establishment process. DH Keys are generated using DH group14 parameters from RFC3526, Section.3.

[RFC3526, Section.3]

3. 2048-bit MODP Group

This group is assigned id 14.

This prime is: 2^2048 - 2^1984 - 1 + 2^64 * { [2^1918 pi] + 124476 }

Its hexadecimal value is:

FFFFFFFF FFFFFFFF C90FDAA2 2168C234 C4C6628B 80DC1CD1 29024E08 8A67CC74 020BBEA6 3B139B22 514A0879 8E3404DD EF9519B3 CD3A431B 302B0A6D F25F1437 4FE1356D 6D51C245 E485B576 625E7EC6 F44C42E9 A637ED6B 0BFF5CB6 F406B7ED EE386BFB 5A899FA5 AE9F2411 7C4B1FE6 49286651 ECE45B3D C2007CB8 A163BF05 98DA4836 1C55D39A 69163FA8 FD24CF5F 83655D23 DCA3AD96 1C62F356 208552BB 9ED52907 7096966D 670C354E 4ABC9804 F1746C08 CA18217C 32905E46 2E36CE3B E39E772C 180E8603 9B2783A2 EC07A28F B5C55DF0 6F4C52C9 DE2BCBF6 95581718 3995497C EA956AE5 15D22618 98FA0510 15728E5A 8AACAA68 FFFFFFFF FFFFFFFF

The generator is: 2.

7.2.3 FCS_CKM.4 Cryptographic Key Destruction

The secret keys stored on flash are cleared by overwriting once with zeros, followed by a read-verify.

The secret keys stored on SDRAM are cleared by overwriting once with a random pattern, followed by a read-verify.

Table 7-1 Key Destructions

Name	Description of Key	Storage	Zeroization	Interface
Diffie-Hellman Shared Secret	This is the shared secret used as part of the Diffie-Hellman key exchange.	SDRAM (plaintext)	Automatically after completion of DH exchange. Overwritten with: zeros.	
Diffie-Hellman private exponent	This is the private exponent used as part of the Diffie-Hellman key exchange.	SDRAM (plaintext)	Automatically after completion of DH exchange. Overwritten with: zeros.	
SSH/TLS session key	The key is used for encrypting/decrypting the traffic in a secure connection.	SDRAM (plaintext)	Automatically after session terminated. Overwritten with: zeros.	Key store API
SSH private host key	The key for authentication.	Internal flash (plaintext)	Overwritten by a command. Overwritten with: a new value of the key.	File system API
TLS private	The key is used for	Internal	Overwritten by a	File system API

Name	Description of Key	Storage	Zeroization	Interface
key	signature and authentication.	flash (plaintext)	command. Overwritten with: a new value of the key.	
Radius shared secret	The key is used for authentication.	Internal flash (plaintext)	Overwritten by a command. Overwritten with: a new value of the key.	File system API
RSA key pair	The key pair is used for digital signature and key establishment.	SDRAM (plaintext)	Automatically after completion of use of the key. Overwritten with: zeros.	Key store API
RSA key pair	The key pair is used for digital signature and key establishment.	CF card (AES256 cipher)	Zeroized using “undo rsa key-pair” command. Overwritten with: zeros.	File system API

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

The TOE provides symmetric encryption and decryption capabilities using AES algorithm with key size 128 bits, 256 bits in GCM mode as specified in ISO 19772.

- AES128 GCM, AES256 GCM are supported by TLS.
- AES128 GCM, AES256 GCM are supported by SSH.

7.2.5 FCS_COP.1/SigGen Cryptographic Operation (Signature Generation and Verification)

The TOE provides cryptographic signature services using RSA with key sizes 2048 bits as specified in FIPS PUB 186-4 “Digital Signature Standard (DSS)”.

- The RSA with key size 2048 bits is used for signature generation and verification of SSH.
- The RSA with key size of 2048 bits is used for signature generation and verification of TLS.

7.2.6 FCS_COP.1/Hash Cryptographic Operation (Hash Algorithm)

The TOE provides cryptographic hashing services using SHA-1, SHA-256, and SHA-384 as specified in FIPS Pub 180-3 “Secure Hash Standard.”, it also meet the ISO/IEC 10118-3:2004.

The association of the hash function with other TSF cryptographic functions:

Table 7-2 Usage of Hash Algorithm

Cryptographic Functions	Hash Function
HMAC-SHA-1	SHA-1
HMAC-SHA2-256	SHA-256
TLS Digital signature verification	SHA-1 SHA-256 SHA-384
SSH Digital signature verification	SHA-1 SHA-256
Hash_DRBG	SHA-256 SHA-384

7.2.7 FCS_COP.1/KeyedHash Cryptographic Operation (Keyed Hash Algorithm)

The TOE provides cryptographic keyed hash services using HMAC-SHA1, HMAC-SHA2-256 according to RFC2104: HMAC, it also complies with the ISO/IEC 9797-2:2011, Section 7 “MAC Algorithm 2”.

Table 7-3 Specification of Keyed Hash Algorithm

HMAC function	Key length (bits)	Hash function	Block size (bits)	Output MAC length (bits)
HMAC-SHA1	160	SHA1	512	160
HMAC-SHA2-256	256	SHA-256	512	256

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

The TOE implements a deterministic random bit generator (DRBG) which is conformant to [ISO18031] using the DRBG mechanism Hash_DRBG as specified in [SP800-90A], chap. 10.1.1.

The entropy source is based on hardware (internal noise source). Random numbers from the internal noise source are only used for seeding the DRBG. The TOE accumulates 256 bits entropy as a seed of deterministic RBG which compy with AIS20 with a deterministic random number generator of class DRG.3.

The TOE set new seed using at least 256 bits entropy before generate random bits as cryptographic key.

7.2.9 FCS_SSHC_EXT.1 SSH Client

7.2.9.1 FCS_SSHC_EXT.1.1

The TOE implements the SSH protocol that comply with RFCs 4251, 4252, 4253, 4254, 5656, 6668.

7.2.9.2 FCS_SSHC_EXT.1.2

Both public key and password authentication modes are supported by SSH client function. Users can use any or both of those modes to login external SSH server successfully.

The supported public key algorithms for authentication include RSA with cryptographic key size of 2048-bit or greater. These public key algorithm conforms to FCS_SSHC_EXT.1.5.

7.2.9.3 FCS_SSHC_EXT.1.3

The max packet length that SSH client can process is 35000 bytes, as defined in RFC4253. If the packet is larger than 35000 bytes, SSH client function will drop this packet.

7.2.9.4 FCS_SSHC_EXT.1.4

The SSH client supports the encryption algorithms of aes128-gcm and aes256-gcm.

When SSH Client establishes a connection, it will send a list of encryption algorithms to SSH server. SSH Server will check each algorithm in the list one by one. If it finds one algorithm in the list that is also supported by it, this algorithm will be chosen as the encryption algorithm between client and server. If no algorithm in the list is supported by SSH server, the connection will be terminated.

After the encryption algorithm is selected, Server and Client will create a random number and exchange. Client and Server will use own random number to create an encryption key.

Then SSH Client will use its own encryption key to encrypt packet, and use SSH Server's encryption key to decrypt packet.

7.2.9.5 FCS_SSHC_EXT.1.5

SSH client function supports the public key algorithm of ssh-rsa.

Before SSHC and SSHS build a connection, they both need to configure a Local Key-pair what is used for authentication. In Huawei device, this local key-pair is used for SSH server and SSH client.

When Client authenticates Server, first step is to consult public key algorithms. Client will send a list of public key algorithms to SSH server. SSH Server will check each algorithm in the list one by one. If it finds one algorithm in the list that is also supported by it, this algorithm will be chosen as the public key algorithm between client and server. If no algorithm in the list is supported by SSH server, the connection will be terminated.

7.2.9.6 FCS_SSHC_EXT.1.6

SSH client supports the data integrity algorithms of hmac-sha1, hmac-sha1-96 and hmac-sha2-256.

7.2.9.7 FCS_SSHC_EXT.1.7

SSH client supports the following key exchange algorithm of diffie-hellman-group14-sha1.

7.2.9.8 FCS_SSHC_EXT.1.8

The SSH connection will be rekeyed after one hour of session time or one gigabyte of transmitted data using that key which ever goes first.

The SSH allows either side to force another run of the key-exchange phase, changing the encryption and integrity keys for the session. The idea is to do this periodically, after one hour of session time or one gigabyte of transmitted data using that key which ever goes first.

7.2.9.9 FCS_SSHC_EXT.1.9

The SSH client will authenticate the identity of the SSH server using a local database associating each host name with its corresponding public key.

7.2.10 FCS_SSHS_EXT.1 SSH Server

7.2.10.1 FCS_SSHS_EXT.1.1

The TOE implements the SSH protocol that complies with RFCs 4251, 4252, 4253, 4254, 5656, and 6668.

7.2.10.2 FCS_SSHS_EXT.1.2

Both public key and password authentication modes are supported by SSH server function. The TOE implements the public key algorithms of ssh-rsa.

SSH users can be authenticated in eight modes: RSA, password, password-RSA, and All (any authentication mode of RSA or password is allowed with “ALL” mode). The SSH user that created by administrators shall configured one of mode. Then the external SSH client can login SSH server successfully via the configured SSH user and authentication mode.

7.2.10.3 FCS_SSHS_EXT.1.3

The max packet length that SSH client can process is 35000 bytes, as defined in RFC 4253. If the packet is larger than 35000 bytes, SSH server function will drop this packet and close the current session.

7.2.10.4 FCS_SSHS_EXT.1.4

SSH server function supports the encryption algorithms of aes128-gcm and aes256-gcm.

When SSH Client establishes a connection, it will send a list of encryption algorithms to SSH server. SSH Server will check each algorithm in the list one by one. If it finds one algorithm in the list that is also supported by it, this algorithm will be chosen as the encryption algorithm between client and server. If no algorithm in the list is supported by SSH server, the connection will be terminated.

After the encryption algorithm is selected, Server and Client will create a random number and exchange. Client and Server will use own random number to create an encryption key.

Then SSH server will use its own encryption key to encrypt packet, and use SSH client’s encryption key to decrypt packet.

7.2.10.5 FCS_SSHS_EXT.1.5

SSH server function supports the public key algorithm of ssh-rsa.

Before SSHC and SSHS build a connection, they both need to configure a Local Key-pair what is used for authentication. In Huawei device, this local key-pair is used for SSH server and SSH client.

When Client authenticates Server, first step is to consult public key algorithms. Client will send a list of public key algorithms to SSH server. SSH Server will check each algorithm in the list one by one. If it finds one algorithm in the list that is also supported by it, this algorithm will be chosen as the public key algorithm between client and server. If no algorithm in the list is supported by SSH server, the connection will be terminated.

7.2.10.6 FCS_SSHS_EXT.1.6

SSH server function supports the data integrity algorithms of hmac-sha1, hmac-sha1-96, and hmac-sha2-256.

7.2.10.7 FCS_SSHS_EXT.1.7

SSH server supports the following key exchange algorithm: diffie-hellman-group14-sha1.

7.2.10.8 FCS_SSHS_EXT.1.8

The SSH connection will be rekeyed after one hour of session time or one gigabyte of transmitted data using that key which ever goes first.

The SSH allows either side to force another run of the key-exchange phase, changing the encryption and integrity keys for the session. The idea is to do this periodically, after one hour of session time or one gigabyte of transmitted data using that key which ever goes first.

7.2.11 FCS_TLSC_EXT.1 Extended: TLS Client Protocol

7.2.11.1 FCS_TLSC_EXT.1.1

The TLS client supports the following ciphersuites:

- TLS_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5288
- TLS_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288

7.2.11.2 FCS_TLSC_EXT.1.2

The TOE supports configuring reference identifier and matching this identifier with server certificate

7.2.11.3 FCS_TLSC_EXT.1.3

Only when the peer certificate is valid the TLS trusted channel can be established. If the peer certificate is invalid, the connection will be rejected.

7.2.11.4 FCS_TLSC_EXT.1.4

TLS don't support EC Extension in the Client Hello.

7.3 Identification and Authentication (FIA)

7.3.1 FIA_AFL.1 Authentication Failure Management

The TOE can be configured within 3 to 5 unsuccessful authentication attempts by Administrators. When the defined number of unsuccessful authentication attempts has been met, the TOE will prevent the offending remote Administrator from successfully authenticating until unlock is taken by a local Administrator or prevent the offending remote Administrator from successfully authenticating until an Administrator defined time period has elapsed.

7.3.2 FIA_PMG_EXT.1 Password Management

The TOE supports the local definition of users with corresponding passwords which are used for security administrators' authentication of local or remote administration connections. The passwords can be composed of any combination of upper and lower case letters, numbers, and special characters (not including spaces or question marks)". Minimum password length is settable by the Authorized Administrator, and support passwords of 15 characters or greater. Password composition rules specifying the types and number of required characters that comprise the password are settable by the Authorized Administrator. Passwords have a maximum lifetime, configurable by the Authorized Administrator.

7.3.3 FIA_UIA_EXT.1 User Identification and Authentication

The TOE requires all users to be successfully identified and authenticated before allowing execution of any TSF mediated action except display of the banner.

The TOE supports user login over console or remote interface. Any login method need authentication before successfully logon.

Local access is achieved by console port. The console interface supports user-based AAA authentication.

Remote access is achieved by SSH. Users can initiate a SSH session to login to a remote interface by user-based AAA authentication. The TOE supports public-key of RSA or username/password for identity authentication. It also supports associated identity authentication of password and public-key. Users can also login with any of the identity authentication modes of password, and RSA when their login mode are configured to be ‘ALL’.

7.3.4 FIA_UAU_EXT.2 Password-based Authentication Mechanism

The TOE can be configured to require local authentication or remote authentication as defined in the authentication policy for interactive (human) users.

The policy for interactive (human) users (Administrators) can be authenticated to the local user database, or have redirection to a remote authentication server. Interfaces can be configured to try one or more remote authentication servers, and then fail back to the local user database if the remote authentication servers are inaccessible.

If the interactive (human) users (Administrators) password is expired, the user is required to create a new password after correctly entering the expired password.

7.3.5 FIA_UAU.7 Protected Authentication Feedback

When a user inputs their password at the local console, the console will not display the input so that the user password is obscured. For remote session authentication, the TOE does not echo any characters as they are entered. The TOE does not provide any additional information to the user that would give any indication about the authentication data.

7.3.6 FIA_X509_EXT.1/Rev X.509 Certificate Validation

The TOE supports to verify the certificate and the certificate path by the rules specified in RFC 5280, using algorithm RSA.

The TOE supports to verify the revocation status by CRLs as specified in RFC 5280.

The TOE validate the certificate by steps as below:

1. Validate basic certificate fields and the extendedKeyUsage field.
2. Validate the revocation status using CRL as specified in [RFC 5759].
3. Validate certificate path as specified in [RFC 5280], do step 1 and 2 for every certificate in the

certificate chain.

4. Validate the end of the certificate chain, it should be trusted root certificate.

7.3.7 FIA_X509_EXT.2 X.509 Certificate Authentication

The certificate used by TLS authentication is sent by TLS server. The CRL should be loaded for certificate validation.

The TOE will send a security log when a connection cannot be established during the validity check of a certificate used in establishing a trusted channel. TLS only supports RSA certificate.

The check of validity of the certificates takes place at authentication of TLS connection and verification of code signing for system software updates. When the certificate is valid, we can trust the peer identity and use the certificate to verify the integrity of the message.

7.4 Security management (FMT)

7.4.1 FMT_MOF.1.1/ManualUpdate The TSF shall restrict the ability to enable the functions to perform manual updates to Security Administrators

Only administrators have the right to create or delete local user. While changing the local user privilege level, the configured new level of the local user cannot be higher than that of the login-in user. In this way no user except administrators can change another user to be at the privilege level of administrator. And only administrators have the ability to perform manual update. So the manual update is restricted to administrators. The TOE uses groups to organize users. Different kinds of users are in different group and every group has a specific level that identity its roles and scope of rights.

7.4.2 FMT_MOF.1/Functions Management of security functions behaviour

Only administrators have right to configure audit servers where audit records are exported to.

7.4.3 FMT_MOF.1/Services Management of security functions behaviour

Only administrators have ability to enable and disable the functions and services, the other users are disallowed to do it.

7.4.4 FMT_MTD.1/CoreData Management of TSF Data

Only administrators have privilege to manage the TSF data, the other users are disallowed to do it.

The TOE provides the ability for authorized administrators to access TOE data, such as audit data, configuration data. Each of the predefined and administratively configured user has different right to access the TOE data.

The access control mechanisms of the TOE are based on hierarchical access levels where a user level is associated with every user and terminal on the one hand and a command level is associated with every command. Only if the user level is equal or higher to a specific command, the user is authorized to execute this command. Management of security function is realized through commands. So for every management function sufficient user level is required for the user to be able to execute the corresponding command.

7.4.5 FMT_MTD.1/CryptoKeys Management of TSF data

Only administrators have the right to delete, generate, import the cryptographic keys, the other users are disallowed to do this.

7.4.6 FMT_SMF.1 Specification of Management Functions

The TOE provides all the capabilities necessary to securely manage the TOE. The administrative user can connect to the TOE using the CLI to perform these functions via SSH encrypted session.

The management functionality provided by the TOE includes the following administrative functions:

- Ability to manage the TOE locally as well as 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 verify the updates are valid
- Ability to configure audit behavior
- Ability to configure thresholds for SSH rekeying

7.4.7 FMT_SMR.2 Restrictions on security roles

A Security Administrator is able to administer the TOE through the local console or through a remote mechanism.

An administrator can create, delete and modify the other users and endow them with a proper right according to the users' roles. The TOE uses groups to organize users. Different kinds of users are in different group and every group has a specific level that identity its roles and scope of rights. Every user in one group has the same scope of rights that the group owns. The TOE has 4 default user groups: manage-ug, system-ug, monitor-ug, and visitor-ug.

7.5 Protection of the TSF (FPT)

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

The TOE stores all pre-shared keys, symmetric keys, and private keys in the file system in Flash that can't be read, copy or extract by administrators; hence no interface access.

7.5.2 FPT_APW_EXT.1 Protection of Administrator Passwords

The administrator passwords are stored to configuration file in cryptographic form hashed with salt by SHA256, including username passwords, authentication passwords, console and virtual terminal line access passwords.

In this manner, the TOE ensures that plaintext user passwords will not be disclosed to anyone through normal interfaces including administrators.

7.5.3 FPT_TST_EXT.1 TSF testing

The TSF run a suite of self-tests during initial start-up to demonstrate the correct operation of the TSF, including software integration verification by CRC check and the correct operation of cryptographic functions. During initial power on start-up, software CRC is checked at first. If CRC check is failed the start-up procedure will stop. After VRP gain control, it test the correct operation of cryptographic functions with known-answer test. If this testing fail the start-up procedure will also stop.

7.5.4 FPT_TUD_EXT.1 Trusted Update

Only authenticated administrators have the ability to manually initiate an update to TOE firmware/software. During the updating procedure, digital signature as defined at FCS_COP.1/SigGen will be verified by the TOE at first.

The administrators can query the currently executing version of the TOE firmware/software as well as the most recently installed version by a command. The currently executing patches and most recently installed patches can also be checked out.

The validation of the firmware/software integrity is always performed before the process of replacing a non-volatile, system resident software component with another is started. All discrete software components (e.g. applications, drivers, kernel, and firmware) of the TSF are archived together into a whole package and the single package is digitally signed. RSADSA as specified in FCS COP.1(2) can be used for firmware/software digital signature mechanism to authenticate it prior to installation and that installation fails if the verification fails.

7.5.5 FPT_STM_EXT.1 Reliable Time Stamps

Only administrators have the ability to modify the time of TOE, and all modification about time will be recorded.

The security functions that make use of time include:

- With this information the real time for all audit data can be calculated.
- The validation period of the certificate can be calculated.

The Network Time Protocol (NTP) is supported by TOE. NTP synchronizes clocks of all devices on a network so that the devices can implement applications based on the uniform time.

NTP is applied in the following situations where all the clocks of hosts or routers in a network need to be consistent:

- Network management: Analysis on logs or debugging information collected from different routers must be performed based on time.
- Charging system: Requires the clocks of all devices to be consistent.
- Completing certain functions: For example, timing restart of all the routers in a network requires the clocks of all the routers to be consistent.

7.6 TOE Access (FTA)

7.6.1 FTA_SSL_EXT.1 TSF-initiated Session Locking

An administrator can configure maximum inactivity times for both local and remote administrative sessions. When a session is inactive (i.e., not session input) for the configured period of time the TOE will terminate the session, flush the screen, and no further activity is allowed requiring the administrator to log in (be successfully identified and authenticated) again to establish a new session.

The allowable range is from 0 minute 0 second to 35791 minutes 59 seconds.

7.6.2 FTA_SSL.3 TSF-initiated Termination

When the remote session is inactive (i.e., not session input) for the configured period of time the TOE will terminate the session.

7.6.3 FTA_SSL.4 User-initiated Termination

When the initiated administrator or local session is inactive (i.e., not session input) for the configured period of time the TOE will terminate the session.

7.6.4 FTA_TAB.1 Default TOE Access Banners

To provide some prompts or alarms to users, Administrator can use the header command to configure a title on the router. If a user logs in to the router, the title is displayed. Administrator can specify the title information, or specify the title information by using the contents of a file. The title displayed same for both local and remote users.

When a terminal (remote or local) connection is activated and attempt to log in, the terminal displays the contents of the title that is set by using the header login command. After the successful login, the terminal displays the contents of the title that is configured by using the header shell command.

The local Console port and the remote Secure-Telnet interface are used for an administrator to communicate with the router.

7.7 Trusted path/channels (FTP)

7.7.1 FTP_ITC.1 Inter-TSF trusted channel

- The TOE protects communications with audit server with TLS.
- The TOE protects communications between a TOE and its connected management server with SSH.
- The TOE protects communications between a TOE and another TOE with SSH.
- TLS/SSH protects the data from disclosure by encryption defined at 6.1.2.6 and ensure that the

data has not been modified by MAC defined by 6.1.2.7 .

7.7.2 FTP_TRP.1 Trusted Path

All remote administrative communications take place over a secure encrypted SSH session. The remote users are able to initiate SSH communications with the TOE.

8 Crypto Disclaimer

The following cryptographic algorithms are used by NE40E to enforce its security policy:

#	Purpose	Cryptographic Mechanism	Standard of Implementation	Key Size in Bits	Standard of Application	Comments
1	Key Generation	RSA schemes	-	2048-bit or greater	FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Appendix B.3	FCS_CKM.1
2	Key Establishment	RSA-based key establishment schemes	Pair-Wise Key Establishment Schemes Using Integer Factorization Cryptography	2048-bit or greater	NIST Special Publication 800-56B	FCS_CKM.2
3	Confidentiality	AES in CTR mode		128 bits or 256 bits	AES as specified in ISO 18033-3, CTR as specified in ISO 10116	FCS_COP.1/ DataEncryption
	Confidentiality	AES in GCM mode		128 bits or 256 bits	AES as specified in ISO 18033-3, GCM as specified in ISO 19772	FCS_COP.1/ DataEncryption

#	Purpose	Cryptographic Mechanism	Standard of Implementation	Key Size in Bits	Standard of Application	Comments
4	Authentication	RSA signature	RSA: PKCS#1_V2.1, RSASSA-PKCS2v1_5	2048 bits	FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Section 5.5	FCS_COP.1/ SigGen
			Digital signature scheme 2 or Digital Signature scheme 3	2048 bits	ISO/IEC 9796-2, Digital signature scheme 2 or Digital Signature scheme 3	FCS_COP.1/ SigGen
	Integrity	HMAC-SHA-1, HMAC-SHA-256	-	160 bits, 256 bits	ISO/IEC 9797-2:2011, Section 7 “MAC Algorithm 2”	FCS_COP.1/Hash
5	Cryptographic Primitive	SHA-1, SHA-256, SHA-384,	-	160 bits, 256 bits, and 384 bits,	ISO/IEC 10118-3:2004	FCS_COP.1/ KeyedHash
6	Random Bit Generation	Hash_DRBG (any); DRG.2 acc. to AIS20	-	256 bits	AIS20 ISO/IEC 18031:2011 Table C.1 “Security Strength Table for Hash Functions”	FCS_RBG.1
7	Trusted Channel	SSH V2.0	RFC 4251 RFC 4252 RFC 4253 RFC 4254 RFC 5656 RFC 6668	-	-	FTP_TRP.1/ Admin
		TLS1.1	RFC 3268 RFC 4346 RFC 5246 RFC 6125	-	-	FTP_ITC.1
		TLS1.2	RFC 3268 RFC 5246 RFC 6125	-	-	FTP_ITC.1
	Cryptographic Primitive	Generation of prime numbers for RSA	None			Miller-Rabin-Test is used as primality test.

Referenced Documents

[AIS20] W. Killmann, W. Schindler; A proposal for: Functionality classes for random number generators, Version 2.0, September 18th 2011.

[FIPS 186-4] National Institute of Standards and Technology, Digital Signature Standard (DSS), Federal Information Processing Standards Publication FIPS PUB 186-4, July 2013

[PKCS#1] RSA Cryptography Specifications Version 2.1(RFC3447)

[PKCS#3] A cryptographic protocol that allows two parties that have no prior knowledge of each other to jointly establish a shared secret key over an insecure communications channel.

[FIPS 198-1]The Keyed-Hash Message Authentication Code (HMAC)--2008 July

[RFC 4251]The Secure Shell (SSH) Protocol Architecture, January 2006

[RFC 4252]The Secure Shell (SSH) Authentication Protocol, January 2006

[RFC 4253]The Secure Shell (SSH) Transport Layer Protocol, January 2006

[RFC 4254]The Secure Shell (SSH) Connection Protocol, January 2006

[RFC 6668]SHA-2 Data Integrity Verification for the Secure Shell (SSH) Transport Layer Protocol

[RFC 3268]Advanced Encryption Standard (AES) Ciphersuites for Transport Layer Security (TLS)

[RFC 4346]The Transport Layer Security (TLS) Protocol Version 1.1

[RFC 5246]The Transport Layer Security (TLS) Protocol Version 1.2

[RFC 6125]Representation and Verification of Domain-Based Application Service Identity within Internet Public Key Infrastructure Using X.509 (PKIX) Certificates in the Context of Transport Layer Security (TLS)

[ANSI X9.31]NIST-Recommended Random Number Generator Based on ANSI X9.31 Appendix A.2.4 Using the 3-Key Triple DES and AES Algorithms

[NIST SP 800-56A]National Institute of Standards and Technology, Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography, May 2013

[NIST SP 800-56B]National Institute of Standards and Technology, Recommendation for Pair-Wise Key Establishment Schemes Using Integer Factorization Cryptography August 2009

[ISO/IEC 18031:2011] Information technology -- Security techniques -- Random bit generation

[ISO 18033-3] Information technology — Security techniques — Encryption algorithms

[ISO/IEC 9796-2]Information technology -- Security techniques -- Digital signature schemes giving message recovery

[ISO/IEC 9797-2]Information technology -- Security techniques -- Message Authentication Codes (MACs)

[ISO/IEC 10118-3]Information technology -- Security techniques -- Hash-functions

[ISO/IEC 14888-3] Information technology -- Security techniques -- Digital signatures with appendix

9 Abbreviations Terminology and References

9.1 Abbreviations

Name	Explanation
AAA	Authentication Authorization Accounting
CA	Certificate Authority
CC	Common Criteria
CEM	Common Evaluation Methodology for Information Technology Security
CLI	Command Line Interface
EAL	Evaluation Assurance Level
EXEC	Execute Command
GUI	Graphical User Interface
IC	Information Center
IP	Internet Protocol
IPC	Inter-Process Communication
LMT	Local Maintenance Terminal
LPU	Line Process Unit
MAN	Metropolitan Area Network
MCU	Main Control Unit
MPU	Main Processing Unit

Name	Explanation
NDcPP	collaborative Protection Profile for Network Device
NE40E	NetEngine40E Universal Service Router
NMS	Network Management Server
NTP	Network Tiem Protocal
PP	Protection Profile
RMT	Remote Maintenance Terminal
SFR	Security Functional Requirement
SFU	Switching Fabric Unit
SPU	Service Process Unit
SRU	Switch Router Unit
SSH	Secure Shell
SSL	Secure Sockets Layer
ST	Security Target
STP	Spanning-Tree Protocol
TLS	Transport Layer Security
TOE	Target of Evaluation
TSF	TOE Security Functions
VRP	Versatile Routing Platform
HPV	Hot Patch Version

9.2 Terminology

This section contains definitions of technical terms that are used with a meaning specific to this document. Terms defined in the [CC] are not reiterated here, unless stated otherwise.

Terminology	Explanation
Administrator:	An administrator is a user of the TOE who may have been assigned specific administrative privileges within the TOE. This ST may use the term administrator occasionally in an informal context, and not in order to refer to a specific role definition – from the TOE's point of view, an administrator is simply a user who is authorized to perform certain administrative actions on the TOE and the objects managed by the TOE. Since all user levels are assigned to commands and users and users can only execute a command if their associated level is equal or higher compared to the level assigned to a command, a user might have certain administrative privileges but lacking some other administrative privileges. So the decision whether a user is also an administrator or not might change with the context (e.g. might be able to change audit settings but cannot perform user management).
Operator:	See User.
User:	A user is a human or a product/application using the TOE which is able to authenticate successfully to the TOE. A user is therefore different to a subject which is just sending traffic through the device without any authentication.

9.3 References

Name	Description
[AIS20]	W. Killmann, W. Schindler; A proposal for: Functionality classes for random number generators, Version 2.0, September 18th 2011.
[CC]	Common Criteria for Information Technology Security Evaluation. Part 1-3 April 2017 Version 3.1 Revision 5 CCMB-2017-001, -002, -003
[CC1]	Common Criteria (CC) Part 1: Introduction and general model April 2017 Version 3.1 Revision 5 CCMB-2017-04-001

Name	Description
[CC2]	Part 2: Security functional components April 2017 Version 3.1 Revision 5 CCMB-2017-04-002
[CC3]	Part 3: Security assurance components April 2017 Version 3.1 Revision 5 CCMB-2017-04-003
[CEM]	Common Methodology for Information Technology Security Evaluation Evaluation methodology April 2017 Version 3.1 Revision 5 CCMB-2017-04-004
[CPP_ND]	collaborative Protection Profile for Network Devices, Version 2.0E, 14-Mar-2018
[C&R]	NDcPP Universal Service Router NE40E V800R010C00SPC200 Configuration and Reference Version: 0.4 Date: 29/09/2018
[PD]	NE40E Product Documentation Procduct Version: V800R010C00SPC200 Library Version: 02 Date: 18/04/2018
[OPE]	Huawei NE40E Series Product V800R010C00SPC200 Operational user Guidance Product Version: V800R010C00SPC200 Version: 0.6 Date: 02/10/2018
[PRE]	Huawei NE40E Series Products V800R010C00SPC200 Preparative Procedures Product Version: V800R010C00SPC200 Version: 0.7 Date: 02/10/2018

Name	Description
[ISO18031]	Information technology — Security techniques — Random bit generation Second edition 2011-11-15
[RFC 3526]	This document defines new Modular Exponential (MODP) Groups for the Internet Key Exchange (IKE) protocol. It documents the well known and used 1536 bit group 5, and also defines new 2048, 3072, 4096, 6144, and 8192 bit Diffie-Hellman groups numbered starting at 14. Please refer to the following link: http://www.rfc-editor.org/info/rfc3526
[RFC 4251]	This document describes the architecture of the SSH protocol, as well as the notation and terminology used in SSH protocol documents. It also discusses the SSH algorithm naming system that allows local extensions. Please refer to the following link: http://www.rfc-editor.org/info/rfc4251
[RFC 5280]	This memo profiles the X.509 v3 certificate and X.509 v2 certificate revocation list (CRL) for use in the Internet. Please refer to the following link: http://www.rfc-editor.org/info/rfc5280
[RFC 5759]	This document specifies a base profile for X.509 v3 Certificates and X.509 v2 Certificate Revocation Lists (CRLs) for use with the United States National Security Agency's Suite B Cryptography. Please refer to the following link: http://www.rfc-editor.org/info/rfc5759
[SD_ND]	Evaluation Activities for Network Device cPP March-2018 Version 2.0+Errata 20180314 from 2018-03-14
[SP800-56A]	Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography Revision 2 May 2013
[SP800-56B]	Recommendation for Pair-Wise Key Establishment Schemes Using Integer Factorization Cryptography Revision 1 September 2014
[SP800-90A]	Recommendation for Random Number Generation Using Deterministic Random Bit Generators Revision 1 June 2015