Nokia 7705 SAR Series with SAR OS 21.10R5 Security Target

Document Version: 1.4

Date: September 28, 2023

2400 Research Blvd
Suite 395
Rockville, MD 20850

Revision History

Version	Date	Changes
Version 1.0	October 24, 2022	First official release.
Version 1.1	January 22, 2023	Updates to the FCS_CKM.1 and FCS_CKM_EXT.2 claims. Minor updates to section 1.6, 5.2.2.8, and 5.2.7.1.
Version 1.2	May 16, 2023	Minor update to FIA_X509_EXT.1/Rev claims in section 6 Updated section 1.9.
Version 1.3	August 11, 2023	Minor updates to section 2.4
Version 1.4	September 28, 2023	Update to Table 5 and a minor update to FCS_CKM.2 on section 6.

Contents

1 Introduction

The Security Target (ST) serves as the basis for the Common Criteria (CC) evaluation and identifies the Target of Evaluation (TOE), the scope of the evaluation, and the assumptions made throughout. This document will also describe the intended operational environment of the TOE, and the functional and assurance requirements that the TOE meets.

1.1 Security Target and TOE Reference

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

Table 1 – TOE/ST Identification

Category	Identifier
ST Title	Nokia 7705 SAR Series with SAR OS 21.10R5 Security Target
ST Version	1.4
ST Date	September 28, 2023
ST Author	Acumen Security, LLC
TOE Identifier	Nokia 7705 SAR Series with SAR OS 21.10R5
TOE Hardware	7705 SAR-18, 7705 SAR-8, 7705 SAR-X, 7705 SAR-H, 7705 SAR-Hc, 7705 SAR-W, 7705 SAR-Wx, and 7705 SAR-Ax
TOE Software	Nokia SAR OS 21.10R5
TOE Developer	Nokia Corporation
Key Words	NDcPP, Non-Distributed Network Device, Service Aggregation Router

1.2 TOE Overview

The TOE is the Nokia 7705 Service Aggregation Router (SAR) series with SAR OS 21.10R5 consisting of the following versions:

- Nokia 7705 SAR-18,
- Nokia 7705 SAR-8,
- Nokia 7705 SAR-X,
- Nokia 7705 SAR-H,
- Nokia 7705 SAR-W,
- Nokia 7705 SAR-Wx,
- Nokia 7705 SAR-Hc, and
- Nokia 7705 SAR-Ax

Versions of the TOE differ in form factor, networking capacity, and processing capacity. Each variant is described in Section 1.4.

The TOE Description section provides an overview of the TOE architecture, including physical boundaries, security functions, and relevant TOE documentation and references.

1.3 TOE Type

The TOE is a network device that is composed of hardware and software and offers a scalable solution to the end users. It satisfies all of the criterion to meet the collaborative Protection Profile for Network Devices, Version 2.2e [NDcPP v2.2e].

1.4 TOE Description

The TOE is a physical, non-distributed network device implementing networking functions essential for service adaptation, aggregation and routing over Ethernet and Internet Protocol routing infrastructure. The primary scenario of deployment is for mobile backhaul, fixed to mobile convergence, mission-critical applications and enterprise applications.

Each variant of the TOE is fully contained in a single chassis. The TOE may interact with external servers to implement the functions and services, and may be administered from a local or remote management station, but neither the servers nor the management stations are parts of the TOE.

The TOE implements a set of security functions and security mechanisms consistent with the requirements set in the collaborative Protection Profile for Network Devices (NDcPP) Version 2.2e. These include security audit, cryptographic algorithms and protocols, authentication of users and peer entities and assigning the users to roles, security management, Protection of the TOE and the TSF, controlling access to the TOE, and trusted channels and paths between the TOE and peer entities and between the TOE and users.

The TOE consists of hardware, software and security guidance documentation. TOE Hardware is contained in the TOE chassis. Variants of the TOE chassis differ in the physical size, precise hardware configuration, the number of network card slots and network interfaces, and throughput capacity. Some variants include network card slots which may be used for configuring the network ports of the product to precisely match the needs of a specific application.

Each variant of the TOE executes identical software, namely, the Nokia Service Aggregation Router Operating System (SAR OS) Release 21.10R5, and is to be used in accordance with a common security guidance. The TOE models are summarized in Table 2.

Table 2 TOE Models

Platform Description	Processors
7705 SAR-18 # of Cores: 8 Frequency: 600 MHz on SAR-18 CSM module	Cavium OCTEON Plus CN5640

Platform Description	Processors
OS: Nokia SAR OS Image Version: 21.10R5	
7705 SAR-8 # of Cores: 6 Frequency: 800 MHz, on CSMv2 module OS: Nokia SAR OS Image Version: 21.10R5	Cavium OCTEON II CN6335
7705 SAR-X # of Cores: 8 Frequency: 800 MHz on chassis OS: Nokia SAR OS Image Version: 21.10R5	Cavium OCTEON II CN6640
7705 SAR-H # of Cores: 2 Frequency: 600 MHz on chassis OS: Nokia SAR OS Image Version: 21.10R5	Cavium OCTEON Plus CN5020
7705 SAR-Hc # of Cores: 2 Frequency: 600 MHz on chassis OS: Nokia SAR OS Image Version: 21.10R5	Cavium OCTEON II CN6020
7705 SAR-W	Cavium OCTEON Plus CN5010

Platform Description	Processors
 # of Cores: 1 Frequency: 500 MHz on chassis OS: Nokia SAR OS Image Version: 21.10R5	
7705 SAR-Wx # of Cores: 2 Frequency: 600 MHz on chassis OS: Nokia SAR OS Image Version: 21.10R5	Cavium OCTEON II CN6020
7705 SAR-Ax # of Cores: 2 Frequency: 600 MHz on chassis OS: Nokia SAR OS Image Version: 21.10R5	Cavium OCTEON II CN6020

The TOE is deployed inside a secure data center or other premises where physical access is effectively controlled. This ensures that only authorized personnel gain physical access to the TOE. Logical access may be through the management station or through the network interfaces. A management station may be local or remote. In addition to the management stations, a CA/CRL Server, AAA Server, Syslog Server and Update Server may be deployed in the same network with the TOE. Access methods to the different management stations and servers are different as are the protocols for protecting network traffic between them and the TOE. The deployment scenario of the TOE is as illustrated in Figure 1.

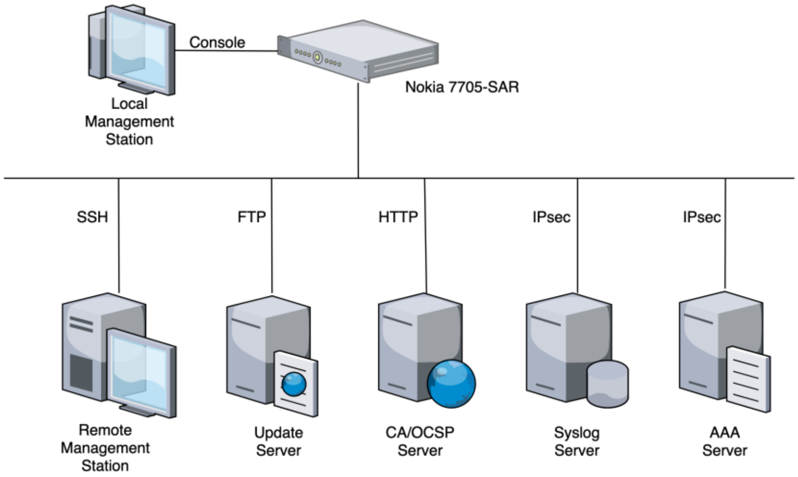

Figure 1 – Representative TOE Deployment

The TOE may be administered locally or remotely. In both methods, the administrative interface is the Command Line Interface (CLI) the TOE software implements for all management functions. If administering the TOE locally, the administrator connects the local management station to the console port of the TOE and operates the TOE in the immediate proximity inside the same data center in which the TOE is deployed. If administering the TOE from a Remote Management Station, the administrator first establishes a Secure Shell (SSH) connection between the Remote Management Station and the TOE, and then proceeds to administer the TOE using the same CLI available to the local administrators from the Local Management Station.

The TOE supports manual upgrading of the TOE software. The administrators load the updates from the developer's web site to a local Update Server or use the developer's update server, then connect the TOE to the Update Server using FTP or SFTP. The software upgrade contains a hash value computed from the image. The hash value shall be verified prior to accepting the upgrade. Therefore, the likelihood of tampering with the TOE software upgrade without detection is very low and there is no need for protection of the connection between the TOE and the Update Server.

To support X.509 certificates with IPsec, the TOE implements a CA/CRL Server used for verifying the certificates and checking their revocation status. As certificates and revocation lists are digitally signed, it is sufficient to connect to them with HTTP. The TOE also may connect to a remote Syslog server for storing audit logs and to an AAA Server storing authentication credentials remotely. Both connections are protected with IPsec.

1.5 TOE Evaluated Configuration

In the evaluated configuration, the TOE consists of the platform as stated in Section 1.4. The TOE supports secure connectivity with another IT environment device as stated in Table 3.

Table 3 – Required Environmental Components

Components	Required (Y/N)	Usage
Local Management Station	Yes	A management station connected to the TOE from the console used for administering the TOE locally.

Components	Required (Y/N)	Usage
Remote Management Station	Yes	A management station connected to the TOE over a network connection, used for administering the TOE remotely over SSH.
SSH Client	Yes	The Remote Management Station must run an SSH client which the remote administrator may use for establishing a secure connection between the Remote Management Station and the TOE.
CA/CRL Server	Yes	A server with a certification authority and certificate revocation list used by the TOE for validating the X.509 certificates used for IKE and IPsec connection establishment.
AAA Server	Yes	A server implementing RADIUS and TACACS+ which the TOE may be configured to use for external authentication of users.
Syslog Server	Yes	A Server to which the TOE may be configured to forward audit log files.
Update Server	Yes	A Server hosting the TOE Software Upgrades. The Administrator may connect to the server and download upgrades for the TOE Software.

1.6 Physical Scope of the TOE

The TOE boundary is the hardware appliance, which is comprised of hardware and software components. It is deployed in an environment that contains the various IT components as depicted in Figure 1. The PDF formats of the TOE guidance documentation is sent to the clients via email by Nokia Sales Team.

1.7 Logical Scope of the TOE

The TOE implements the following security functional requirements:

- Security Audit
- Cryptographic Support
- Identification and Authentication
- Security Management
- Protection of the TSF
- TOE Access
- Trusted Path/Channels

Each of these security functionalities are listed in more detail in the sections below.

1.7.1 Security Audit

The TOE generates audit events for all start-up and shut-down functions and all auditable events as specified in Table 15. Audit events are also generated for management actions specified in FAU_GEN.1. The TOE is capable of storing audit events locally and exporting them to an external audit server over IPsec protocol. Each audit record contains the date and time of event, type of event, subject identity, and the relevant data of the event. The audit server supports the following severity levels: indeterminate (info), major, and minor.

1.7.2 Cryptographic Support

The TOE provides cryptographic support for the services described in Table 4.

Table 4 TOE Cryptography Implementation

Cryptographic Method	Usage
FCS_CKM.1 Cryptographic Key Generation	Cryptographic key generation conforming to the following: • RSA schemes that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.3, • FFC Schemes using 'safe-prime' groups that meet the following: "NIST Special Publication 800-56A Revision 3, Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" and RFC 3526. RSA Key sizes supported are 2048 bits
FCS_CKM.2 Cryptographic Key Establishment	RSA-based key establishment schemes that meet the following: • RSA-based key establishment schemes that meet the RSAES-PKCS1-v1_5 as specified in Section 7.2 of RFC 3447, "Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1", and • FFC Schemes using "safe-prime" groups that meet the 'NIST Special Publication 800-56A Revision 3, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" and RFC 3526.
FCS_CKM.4 Cryptographic Key Destruction	Refer to Table 17 for Key Zeroization details.
FCS_COP.1/DataEncryption	AES encryption and decryption conforming to CBC as specified in ISO 10116 and CTR as specified in ISO 10116. AES key size supported is 128 bits, 192 bits and 256 bits AES modes supported are CBC and CTR.
FCS_COP.1/SigGen	RSA digital signature algorithm conforming to 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. RSA keys are generated by the TOE and are of the size of 2048 bits.
FCS_COP.1/Hash	Cryptographic hashing services conforming to ISO/IEC 10118-3:2004. Hashing algorithms supported are SHA-1, SHA-256, SHA-384 and SHA-512. Message digest sizes supported are: 160, 256, 384 and 512 bits.

Cryptographic Method	Usage
FCS_COP.1/KeyedHash	Keyed-hash message authentication conforming to ISO/IEC 9797-2:2011, Section 7 “MAC Algorithm 2. Keyed-hash algorithm supported are HMAC-SHA1, HMAC-SHA-256, HMAC-SHA-384 and HMAC-SHA-512. Key sizes supported are: 160, 256, 384, and 512 bits. Message digest sizes supported are: 160, 256, 384 and 512 bits.
FCS_RBG_EXT.1 Random Bit Generation	Random number generation conforming to ISO/IEC 18031:2011. The TOE leverages CTR_DRBG(AES). CTR_DRBG seeded with a minimum of 256 bits of entropy.
FCS_IPSEC_EXT.1 IPsec Protocol	The TOE implements the IPsec architecture as specified in RFC 4301. Only tunnel mode is implemented. IPsec ESP is implemented in accordance with RFC4303 using AES in CBC mode with a key of 128 bits, 192 bits and 256 bits. The HMAC algorithms implemented are HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384 and HMAC-SHA-512. IPsec key exchange is implemented with the following: • IKEv1 using Main Mode for Phase 1 exchanges in accordance with RFCs 2407, 2408, 2409, 4109 and 4868. • IKEv2 in accordance with RFC 5996 and RFC 4868 as well as with mandatory support for NAT traversal as specified in RFC 5996, section 2.23. IKE payload is encrypted with AES in CBC mode using a 128-bit, 192-bit and 256-bit in accordance with RFC 3602. Security Association life-times are configurable by the Administrator. IKE Protocols perform peer authentication using RSA with either X.509v3 certificates in conformance to RFC 4945 or using pre-shared keys. The secret exponents are generated by the random bit generator implemented in the TOE for Diffie-Hellman groups 14 and 15. Nonces used by IKEv1 and IKE v2 are generated in accordance with the security strength of the negotiated Diffie-Hellman group.

Cryptographic Method	Usage
FCS_SSHS_EXT.1 SSH Server Protocol	The TOE implements SSH v2 protocol in accordance with the following RFCs: 4251, 4252, 4253, 4254, 4256, 4344, 6668, 8268, 8308 (Section3.1) and 8332. The TOE supports public key and password-based authentication. Packets greater than 65000 bytes in an SSH transport connection are dropped. SSH public-key authentication uses ssh-rsa. SSH transport uses the following encryption algorithms: aes128-ctr, aes128-cbc, aes256-cbc and aes256-ctr. SSH transport uses the following data integrity MAC algorithms: hmac-sha1, hmac-sha256, and hmac-sha2-512. Key exchange algorithms supported are diffie-hellman-group14-sha256, diffie-hellman-group14- sha1 and diffie-hellman-group16-sha512. The TOE ensures 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.

The CAVP certificate numbers for the cryptographic algorithms are given in Table 5 The TOE uses Nokia SAR OS 21.10R5 with OpenSSL v1.1.1g, Winpath 3 or Winpath 4 to implement the protocol logic as well as all the cryptographic primitives used by the protocols.

Table 5 CAVP Details

SFR	Algorithm in ST	Implementation name	Operational Environment	CAVP Alg.	CAVP Cert #
FCS_CKM.1	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	Nokia 7705 SAR OS Cryptographic library	Cavium OCTEON II CN6335 Cavium OCTEON II CN6640 Cavium OCTEON II CN6020	RSA KeyGen (FIPS186-4)	C2023
		Nokia 7705 SAR OS Cryptographic library	Cavium OCTEON Plus CN5640 Cavium OCTEON Plus CN5020	RSA KeyGen (FIPS186-4)	C2024

SFR	Algorithm in ST	Implementation name	Operational Environment	CAVP Alg.	CAVP Cert #
			Cavium OCTEON Plus CN5010		
	FFC Schemes using 'safe-prime' groups that meet the following: "NIST Special Publication 800-56A Revision 3, Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" and [RFC 3526]	Nokia 7705 SAR OS Cryptographic library	Cavium OCTEON II CN6335 Cavium OCTEON II CN6640 Cavium OCTEON II CN6020	Safe Primes Key Generation Safe Primes Key Verification	A3133
		Nokia 7705 SAR OS Cryptographic library	Cavium OCTEON Plus CN5640 Cavium OCTEON Plus CN5020 Cavium OCTEON Plus CN5010	Safe Primes Key Generation Safe Primes Key Verification	A3134
FCS_CKM.2	RSA-based key establishment schemes that meet the following: RSAES-PKCS1-v1_5 as specified in Section 7.2 of RFC 3447, "Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1"	Nokia 7705 SAR OS Cryptographic library	Cavium OCTEON II CN6335 Cavium OCTEON II CN6640 Cavium OCTEON II CN6020 Cavium OCTEON Plus CN5640 Cavium OCTEON Plus CN5020 Cavium OCTEON Plus CN5010	None	CCTL tested as per the PP/SD Evaluation Activities

SFR	Algorithm in ST	Implementation name	Operational Environment	CAVP Alg.	CAVP Cert #
	FFC Schemes using "safe-prime" groups that meet the following: 'NIST Special Publication 800-56A Revision 3, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" and [RFC 3526]	Nokia 7705 SAR OS Cryptographic library	Cavium OCTEON Plus CN5640 Cavium OCTEON II CN6640 Cavium OCTEON Plus CN5010	KAS-FFC-SSC Sp800-56Ar3	A3133
		Nokia 7705 SAR OS Cryptographic library	Cavium OCTEON II CN6335 Cavium OCTEON II CN6640 Cavium OCTEON II CN6020	KAS-FFC-SSC Sp800-56Ar3	A3134
FCS_COP.1/ DataEncryption	AES used in [CBC, CTR] mode and cryptographic key sizes [128 bits, 192 bits, 256 bits]	Nokia 7705 SAR OS Cryptographic library	Cavium OCTEON II CN6335 Cavium OCTEON II CN6640 Cavium OCTEON II CN6020	AES-CBC AES-CTR	C2023
		Nokia 7705 SAR OS Cryptographic library	Cavium OCTEON Plus CN5640 Cavium OCTEON Plus CN5020 Cavium OCTEON Plus CN5010	AES-CBC AES-CTR	C2024
FCS_COP.1/ SigGen	For RSA schemes: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 5.5, using PKCS #1 v2.1 Signature Schemes RSASSA-PSS	Nokia 7705 SAR OS Cryptographic library	Cavium OCTEON II CN6335 Cavium OCTEON II CN6640	RSA SigGen (FIPS186-4) RSA SigVer (FIPS186-4)	C2023

SFR	Algorithm in ST	Implementation name	Operational Environment	CAVP Alg.	CAVP Cert #
	and/or RSASSA-PKCS1v1_5; ISO/IEC 9796-2, Digital signature scheme 2 or Digital Signature scheme 3		Cavium OCTEON II CN6020		
		Nokia 7705 SAR OS Cryptographic library	Cavium OCTEON Plus CN5640 Cavium OCTEON Plus CN5020 Cavium OCTEON Plus CN5010	RSA SigGen (FIPS186-4) RSA SigVer (FIPS186-4)	C2024
FCS_COP.1/ Hash	[SHA-1, SHA-256, SHA-384, SHA-512] and message digest sizes [160, 256, 384, 512] bits	Nokia 7705 SAR OS Cryptographic library	Cavium OCTEON II CN6335 Cavium OCTEON II CN6640 Cavium OCTEON II CN6020	SHS	C2023
		Nokia 7705 SAR OS Cryptographic library	Cavium OCTEON Plus CN5640 Cavium OCTEON Plus CN5020 Cavium OCTEON Plus CN5010	SHS	C2024
FCS_COP.1/ KeyedHash	[HMAC-SHA-1, HMAC-SHA- 256, HMAC-SHA-384, HMAC-SHA-512] and cryptographic key sizes [160-bits, 256-bits, 384-bits, 512-bits] and message digest sizes [160, 256, 384, 512] bits	Nokia 7705 SAR OS Cryptographic library	Cavium OCTEON II CN6335 Cavium OCTEON II CN6640 Cavium OCTEON II CN6020	HMAC-SHA-1 HMAC-SHA-256 HMAC-SHA-384 HMAC-SHA-512	C2023

SFR	Algorithm in ST	Implementation name	Operational Environment	CAVP Alg.	CAVP Cert #
		Nokia 7705 SAR OS Cryptographic library	Cavium OCTEON Plus CN5640 Cavium OCTEON Plus CN5020 Cavium OCTEON Plus CN5010	HMAC-SHA-1 HMAC-SHA-256 HMAC-SHA-384 HMAC-SHA-512	C2024
FCS_RBG_EXT.1	CTR_DRBG (AES)	Nokia 7705 SAR OS Cryptographic library	Cavium OCTEON II CN6335 Cavium OCTEON II CN6640 Cavium OCTEON II CN6020	Counter DRBG	C2023
		Nokia 7705 SAR OS Cryptographic library	Cavium OCTEON Plus CN5640 Cavium OCTEON Plus CN5020 Cavium OCTEON Plus CN5010	Counter DRBG	C2024

1.7.3 Identification and Authentication

The TOE supports Role Based Access Control. All users must be authenticated to the TOE prior to carrying out any management actions. The TOE supports password-based authentication and public key-based authentication. Based on the assigned role, a user is granted a set of privileges to access the system.

1.7.4 Security Management

The TOE supports local and remote management of its security functions including:

- Local console CLI administration
- Remote CLI administration via SSHv2
- Timed user lockout after multiple failed authentication attempts

- Password configurations
- Role Based Access Control
- Configurable banners to be displayed at login
- Timeouts to terminate administrative sessions after a set period of inactivity
- Protection of secret keys and passwords

1.7.5 TOE Access

Prior to establishing an administration session with the TOE, a banner is displayed to the user. The banner messaging is customizable. The TOE will terminate an interactive session after configurable number of minutes of session inactivity. A user can terminate their local CLI session and remote CLI session by entering the appropriate command at the prompt.

1.7.6 Protection of the TSF

The TOE protects all passwords, pre-shared keys, symmetric keys, and private keys from unauthorized disclosure. Pre-shared keys, symmetric keys, and private keys are stored in encrypted format. Passwords are stored as a non-reversible hash value as per standard Linux approach. The TOE executes self-tests during initial start-up to ensure correct operation and enforcement of its security functions. An administrator can install software updates to the TOE. The TOE internally maintains the date and time.

1.7.7 Trusted Path/Channels

The TOE supports IPsec for secure communication to the audit server and with the authentication server. The termination points of the IPsec are the TOE and another IPsec implementation. The TOE supports local CLI and uses SSH v2 for secure remote administration.

1.8 Product Functionality not Included in the Scope of the Evaluation

The following product functionality is not included in the CC evaluation:

- FTP and Telnet and are disabled.
- NTP is not used.
- TACACS+ cryptographic protection of the sessions is not covered by the evaluation but the security of TACACS+ relies on IPsec between the TOE and the AAA Server.
- MPLS and SNMP are not included in the scope of the evaluation.
- MACsec functionality is not supported.

1.9 TOE Documentation

Table 6 lists the TOE guidance documentation. The Common Criteria (CC) guidance document and TOE Security Target (ST) are provided in .pdf form on the NIAP portal.

Table 6 TOE Documentation

Reference	Title	Version	Date
[CC]	NOKIA 7705 SERVICE AGGREGATION ROUTER | RELEASE 21.10R5 Common Criteria Admin Guide	1.0	May 15, 2023

Reference	Title	Version	Date
[AGD_1]	NOKIA 7705 SERVICE AGGREGATION ROUTER | RELEASE 21.10R1 Basic System Configuration Guide	01	October 2021
[AGD_2]	NOKIA 7705 SERVICE AGGREGATION ROUTER | RELEASE 21.10R1 Interface Configuration Guide	01	October 2021
[AGD_3]	NOKIA 7705 SERVICE AGGREGATION ROUTER | RELEASE 21.10R1 Log Events Guide	01	October 2021
[AGD_4]	NOKIA 7705 SERVICE AGGREGATION ROUTER | RELEASE 21.10R1 Router Configuration Guide	01	October 2021
[AGD_5]	NOKIA 7705 SERVICE AGGREGATION ROUTER | RELEASE 21.10R1 Services Guide	01	October 2021
[AGD_6]	NOKIA 7705 SERVICE AGGREGATION ROUTER | RELEASE 21.10R1 System Management Guide	01	October 2021
[ST]	Nokia 7705 SAR Series with SAR OS 21.10R5 Security Target	1.3	September 28, 2023

1.10 Other References

In addition to the TOE documentation, the following references are applied within this ST:

- Collaborative Protection Profile for Network Devices, Version 2.2e [NDcPP v2.2e]

2 Conformance Claims

This section identifies the TOE conformance claims, conformance rationale, and relevant Technical Decisions (TDs).

2.1 CC Conformance Claims

The TOE is conformant to the following:

- Common Criteria for Information Technology Security Evaluations Part 1, Version 3.1, Revision 5, April 2017
- Common Criteria for Information Technology Security Evaluations Part 2, Version 3.1, Revision 5, April 2017 (Extended)
- Common Criteria for Information Technology Security Evaluations Part 3, Version 3.1, Revision 5, April 2017 **(**Conformant)

2.2 Protection Profile Conformance

This ST claims exact conformance to the following:

- collaborative Protection Profile for Network Devices, Version 2.2e, 27 March 2020 [NDcPP]

2.3 Conformance Rationale

This ST provides exact conformance to the items listed in the previous section. The security problem definition, security objectives, and security requirements in this ST are all taken from the Protection Profile (PP), performing only the operations defined there.

2.4 Technical Decisions

All NIAP TDs issued to date and applicable to NDcPP v2.2e have been considered. Table 7 identifies all applicable TDs.

Table 7 – Relevant Technical Decisions

Technical Decision	Applicable (Y/N)	Exclusion Rationale (if applicable)
TD0527: Updated to Certificate Revocation Testing (FIA_X509_EXT.1)	Y	
TD0528: NIT Technical Decision for Missing EAs for FCS_NTP_EXT.1.4	N	The ST does not claim FCS_NTP_EXT.1.
TD0536: NIT Technical Decision for Update Verification Inconsistency	Y	
TD0537: NIT Technical Decision for Incorrect reference to FCS_TLSC_EXT.2.3	N	The ST does not claim FCS_TLSC_EXT.2.3
TD0546: NIT Technical Decision for DTLS – clarification of Application Note 63	N	The ST does not claim FCS_DTLSC_EXT.
TD0547: NIT Technical Decision for Clarification on developer disclosure of AVA_VAN	Y	

Technical Decision	Applicable (Y/N)	Exclusion Rationale (if applicable)
TD0555: NIT Technical Decision for RFC Reference incorrect in TLSS Test	N	The ST does not claim FCS_TLSS_EXT.
TD0556: NIT Technical Decisions for RFC 5077 question	N	The ST does not claim FCS_TLSS_EXT.
TD0563: NIT Technical Decision for Clarification of audit date information	Y	
TD0564: NIT Technical Decision for Vulnerability Analysis Search Criteria	Y	
TD0569: NIT Technical Decision for Session ID Usage Conflict in FCS_DTLSS_EXT.1.7	N	The ST does not claim FCS_DTLSS_EXT.
TD0570: NIT Technical Decision for Clarification about FIA_AFL.1	Y	
TD0571: NIT Technical Decision for Guidance on how to handle FIA_AFL.1	Y	
TD0572: NIT Technical Decision for Restricting FTP_ITC.1 to only IP address identifiers	N	The ST does not claim FCS_DTLSC_EXT or FCS_TLSC_EXT.
TD0580: NIT Technical Decision for clarification about use of DH14 in NDcPPv2.2e	Y	
TD0581: NIT Technical Decision for Elliptic curve-based key establishment and NIST SP 800-56Arev3	N	The TOE does not implement elliptic curve cryptography
TD0591: NIT Technical Decision for Virtual TOEs and hypervisors	N	The TOE is not a virtual TOE.
TD0592: NIT Technical Decision for Local Storage of Audit Records	N	The ST does not claim FAU_STG.1.
TD0631: NIT Technical Decision for Clarification of public key authentication for SSH Server	Y	
TD0632: NIT Technical Decision for Consistency with Time Data for vNDs	N	The TOE is not a vND.
TD0633: NIT Technical Decision for IPsec IKE/SA Lifetimes Tolerance	Y	
TD0634: NIT Technical Decision for Clarification required for testing IPv6	N	IPv6 is not supported by the TOE in the evaluated configuration.
TD0635: NIT Technical Decision for TLS Server and Key Agreement Parameters	N	The ST does not claim FCS_TLSS_EXT.1
TD0636: NIT Technical Decision for Clarification of Public Key User Authentication for SSH	N	The ST does not claim FCS_SSHC_EXT.1
TD0638: NIT Technical Decision for Key Pair Generation for Authentication	Y	
TD0639: NIT Technical Decision for Clarification for NTP MAC Keys	N	The ST does not claim FCS_NTP_EXT.1.
TD0670: NIT Technical Decision for Mutual and Non-Mutual Auth TLSC Testing	N	The TOE does not support TLS mutual authentication.

Technical Decision	Applicable (Y/N)	Exclusion Rationale (if applicable)
TD0738: NIT Technical Decision for Link to Allowed-With List	Y	

3 Security Problem Definition

The security problem definition has been taken directly from the claimed PP specified in Section 2.2 and is reproduced here for the convenience of the reader. The security problem is described in terms of the threats that the TOE is expected to address, assumptions about the operational environment, and any Organizational Security Policies (OSPs) that the TOE is expected to enforce.

3.1 Threats

The threats included in Table 8 are drawn directly from the NDcPP2.2E specified in Section 2.2.

Table 8 – Threats

ID	Threat
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.
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.
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.
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

ID	Threat
	and integrity, and potentially the Network Device itself could be compromised.
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.
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.
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.
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.
T.SECURITY_FUNCTIONALITY_FAILURE	An external, unauthorized entity could make use of failed or compromised security functionality and might therefore subsequently use or abuse security functions without prior authentication to access, change or modify device data, critical network traffic or security functionality of the device.

3.2 Assumptions

The assumptions included in Table 9 are drawn directly from NDcPPv2.2E.

Table 9 – Assumptions

ID	Assumption
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 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 cPP does not include any requirements on physical tamper protection or other physical attack mitigations. The cPP does 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. For vNDs, this assumption applies to the physical platform on which the VM runs.
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).
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 the ND cPP. It is assumed that this protection will be covered by cPPs and PP-Modules for particular types of Network Devices (e.g., firewall).
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 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. For TOEs supporting X.509v3 certificate-based authentication, the Security Administrator(s) are expected to fully validate (e.g. offline verification) any CA certificate (root CA certificate or intermediate CA certificate) loaded into the TOE's trust store (aka 'root store', ' trusted CA Key Store', or similar) as a trust anchor prior to use (e.g. offline verification).
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.
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.

ID	Assumption
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.

3.3 Organizational Security Policies

The OSPs included in Table 10 are drawn directly from the NDcPPv2.2E.

Table 10 – OSPs

ID	OSP
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.

4 Security Objectives

The security objectives have been taken directly from the NDcPPv2.2E and are reproduced here for the convenience of the reader.

NDcPP does not state any security objectives for the TOE.

Security objectives rationale is stated in NDcPP and is not reproduced here.

4.1 Security Objectives for the Operational Environment

Security objectives for the operational environment assist the TOE in correctly providing its security functionality. These objectives, which are found in the table below, track with the assumptions about the TOE operational environment.

Table 11 – Security Objectives for the Operational Environment

ID	Objectives for the Operational Environment
OE.PHYSICAL	Physical security, commensurate with the value of the TOE and the data it contains, is provided by the environment.
OE.NO_GENERAL_PURPOSE	There are no general-purpose computing capabilities (e.g., compilers or user applications) available on the TOE, other than those services necessary for the operation, administration and support of the TOE.
OE.NO_THRU_TRAFFIC_PROTECTION	The TOE does not provide any protection of traffic that traverses it. It is assumed that protection of this traffic will be covered by other security and assurance measures in the operational environment.
OE.TRUSTED_ADMN	Security Administrators are trusted to follow and apply all guidance documentation in a trusted manner. For TOEs supporting X.509v3 certificate-based authentication, the Security Administrator(s) are assumed to monitor the revocation status of all certificates in the TOE's trust store and to remove any certificate from the TOE’s trust store in case such certificate can no longer be trusted.
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.
OE.ADMIN_CREDENTIALS_SECURE	The Administrator’s credentials (private key) used to access the TOE must be protected on any other platform on which they reside.
OE.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 Security Requirements

This section identifies the Security Functional Requirements (SFRs) for the TOE. The SFRs included in this section are derived from Part 2 of the Common Criteria for Information Technology Security Evaluation, Version 3.1, Revisions 5, September 2017, NDcPP and all international interpretations.

Table 12 – SFRs

Requirement	Description
FAU_GEN.1	Audit Data Generation
FAU_GEN.2	User Identity Association
FAU_STG_EXT.1	Protected Audit Event Storage
FCS_CKM.1	Cryptographic Key Generation
FCS_CKM.2	Cryptographic Key Establishment
FCS_CKM.4	Cryptographic Key Destruction
FCS_COP.1/DataEncryption	Cryptographic Operation (AES Data Encryption/Decryption)
FCS_COP.1/SigGen	Cryptographic Operation (Signature Generation and Verification)
FCS_COP.1/Hash	Cryptographic Operation (Hash Algorithm)
FCS_COP.1/KeyedHash	Cryptographic Operation (Keyed Hash Algorithm)
FCS_IPSEC_EXT.1	IPsec Protocol
FCS_RBG_EXT.1	Random Bit Generation
FCS_SSHS_EXT.1	SSH Server Protocol
FIA_AFL.1	Authentication Failure Management
FIA_PMG_EXT.1	Password Management
FIA_UIA_EXT.1	User Identification and Authentication
FIA_UAU_EXT.2	Password-based Authentication Mechanism
FIA_UAU.7	Protected Authentication Feedback
FIA_X509_EXT.1/Rev	X.509 Certificate Validation
FIA_X509_EXT.2	X.509 Certificate Authentication
FIA_X509_EXT.3	X.509 Certificate Requests
FMT_MOF.1/Functions	Management of Security Functions Behaviour
FMT_MOF.1/ManualUpdate	Management of Security Functions Behaviour
FMT_MOF.1/Services	Management of Security Functions Behaviour
FMT_MTD.1/CoreData	Management of TSF Data
FMT_MTD.1/CryptoKeys	Management of TSF Data
FMT_SMF.1	Specification of Management Functions
FMT_SMR.2	Restrictions on security roles
FPT_SKP_EXT.1	Protection of TSF Data (for reading of all pre-shared, symmetric and private keys)
FPT_APW_EXT.1	Protection of Administrator Passwords
FPT_TST_EXT.1	TSF Testing

Requirement	Description
FPT_STM_EXT.1	Reliable Time Stamps
FPT_TUD_EXT.1	Trusted Update
FTA_SSL.3	TSF-initiated Termination
FTA_SSL.4	User-initiated Termination
FTA_SSL_EXT.1	TSF-initiated Session Locking
FTA_TAB.1	Default TOE Access Banner
FTP_ITC.1	Inter-TSF Trusted Channel
FTP_TRP.1/Admin	Trusted Path

5.1 Conventions

The CC allows the following types of operations to be performed on the functional requirements: assignments, selections, refinements, and iterations. The following font conventions are used within this document to identify operations defined by CC:

- Assignment: Indicated with *italicized* text;
- Refinement: Indicated with **bold** text;
- Selection: Indicated with <u>underlined</u> text;
- Iteration: Indicated by appending the iteration identifier after a slash, e.g., /SigGen.
- Where operations were completed in the PP and relevant EPs/Modules/Packages, the formatting used in the PP has been retained.
- Extended SFRs are identified by the addition of “EXT” after the requirement name.

5.2 Security Functional Requirements

This section includes the security functional requirements for this ST.

5.2.1 Security Audit (FAU)

5.2.1.1 FAU_GEN.1 Audit Data Generation

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) Auditable events for the <u>not specified</u> 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).*
    - *[<u>no other actions</u>];*

d) *Specifically defined auditable events listed in* Table 13.

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 cPP/ST, *information specified in column three of* Table 13.

Table 13 – Security Functional Requirements and Auditable Events

Requirement	Auditable Events	Additional Audit Record Contents
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_IPSEC_EXT.1	Failure to establish an IPsec SA.	Reason for failure
FCS_RBG_EXT.1	None.	None.
FCS_SSHS_EXT.1	Failure to establish an SSH session	Reason for failure
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 identification and authentication mechanism	Origin of the attempt (e.g., IP address)
FIA_UAU_EXT.2	All use of identification and authentication mechanism	Origin of the attempt (e.g., IP address)
FIA_UAU.7	None.	None.
FIA_X509_EXT.1/Rev	Unsuccessful attempt to validate a certificate Any addition, replacement or removal of trust anchors in the TOE's trust store	Reason for failure of certificate validation Identification of certificates added, replaced or removed as trust anchor in the TOE's trust store
FIA_X509_EXT.2	None.	None.
FIA_X509_EXT.3	None.	None.
FMT_MOF.1/Functions	None.	None.
FMT_MOF.1/ManualUpdate	Any attempt to initiate a manual update	None.
FMT_MOF.1/Services	None.	None.

Requirement	Auditable Events	Additional Audit Record Contents
FMT_MTD.1/CoreData	None.	None.
FMT_MTD.1/CryptoKeys	None.	None.
FMT_SMF.1	All management activities of TSF data	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_STM_EXT.1	Discontinuous changes to time - either Administrator actuated or changed via an automated process	For discontinuous changes to time: The old and new values for the time. Origin of the attempt to change time for success and failure (e.g., IP address).
FPT_TUD_EXT.1	Initiation of update; result of the update attempt (success or failure)	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_SSL_EXT.1	The termination of a local session by the session locking mechanism	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. Failure of the trusted path functions.	None.

5.2.1.2 FAU_GEN.2 User Identity Association

FAU_GEN.2.1

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

5.2.1.3 FAU_STG_EXT.1 Protected Audit Event Storage

FAU_STG_EXT.1.1

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

FAU_STG_EXT.1.2

The TSF Shall be able to store generated audit data on the TOE itself. In addition [

- *The TOE shall consist of a single standalone component that stores audit data locally,*

].

FAU_STG_EXT.1.3

The TSF shall [*overwrite previous audit records according to the following rule: [overwrite the oldest logs]*] when the local storage space for audit data is full.

5.2.2 Cryptographic Support (FCS)

5.2.2.1 FCS_CKM.1 Cryptographic Key Generation

FCS_CKM.1.1

The TSF shall generate **asymmetric** cryptographic key 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;*
- *FFC Schemes using 'safe-prime' groups that meet the following: "NIST Special Publication 800-56A Revision 3, Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" and [RFC 3526].*

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

5.2.2.2 FCS_CKM.2 Cryptographic Key Establishment

FCS_CKM.2.1

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

- *RSA-based key establishment schemes that meet the following: RSAES-PKCS1-v1_5 as specified in Section 7.2 of RFC 3447, "Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1";*
- *FFC Schemes using "safe-prime" groups that meet the following: 'NIST Special Publication 800-56A Revision 3, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" and [RFC 3526].*

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

5.2.2.3 FCS_CKM.4 Cryptographic Key Destruction

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 [*
    - *instructs a part of the TSF to destroy the abstraction that represents the key*]

that meets the following: *No Standard*

5.2.2.4 FCS_COP.1/DataEncryption Cryptographic Operations (AES Data Encryption/Decryption)

FCS_COP.1.1/DataEncryption

The TSF shall perform *encryption/decryption* in accordance with a specified cryptographic algorithm *AES used in* [*CBC, CTR*] *mode* and cryptographic key sizes [*128 bits, 192 bits, 256 bits*] that meet the following: *AES as specified in ISO 18033-3,* [*CBC as specified in ISO 10116, CTR as specified in ISO 10116*].

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

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,*

].

5.2.2.6 FCS_COP.1/Hash Cryptographic Operations (Hash Algorithm)

FCS_COP.1.1/Hash

The TSF shall perform *cryptographic hashing services* in accordance with a specified cryptographic algorithm [*SHA-1, SHA-256, SHA-384, SHA-512*] ~~and cryptographic key sizes [*assignment: cryptographic key sizes*~~] and **message digest sizes [*160, 256, 384, 512*] bits** that meet the following: *ISO/IEC 10118-3:2004*.

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

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, HMAC-SHA-384, HMAC-SHA-512*] and cryptographic key sizes *[160-bits, 256-bits, 384-bits, 512-bits]* **and message digest sizes [*160, 256, 384, 512*] bits** that meet the following: *ISO/IEC 9797-2:2011, Section 7 "MAC Algorithm 2"*.

5.2.2.8 FCS_IPSEC_EXT.1 IPSec Protocol

FCS_IPSEC_EXT.1.1

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

FCS_IPSEC_EXT.1.2

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

FCS_IPSEC_EXT.1.3

The TSF shall implement [*tunnel mode*].

FCS_IPSEC_EXT.1.4

The TSF shall implement the IPsec protocol ESP as defined by RFC 4303 using the cryptographic algorithms [*AES-CBC-128 (RFC 3602), AES-CBC-192 (RFC 3602), AES-CBC-256 (RFC 3602)*] together with a Secure Hash Algorithm (SHA)-based HMAC [*HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512].*

FCS_IPSEC_EXT.1.5

The TSF shall implement the protocol: [

- *IKEv1, using Main Mode for Phase 1 exchanges, as defined in RFCs 2407, 2408, 2409, RFC 4109,* [*no other RFCs for extended sequence numbers*]*, and* [*RFC 4868 for hashfunctions*]*;*
- *IKEv2 as defined in RFC 5996 and* [*with mandatory support for NAT traversal as specified in RFC 5996, section 2.23)*]*, and* [*RFC 4868 for hash functions*]

].

FCS_IPSEC_EXT.1.6

The TSF shall ensure the encrypted payload in the [*IKEv1*, *IKEv2*] protocol uses the cryptographic algorithms [*AES-CBC-128, AES-CBC-192, AES-CBC-256 (specified in RFC 3602)*].

FCS_IPSEC_EXT.1.7

The TSF shall ensure that [

- *IKEv1 Phase 1 SA lifetimes can be configured by a Security Administrator based on* [
  - *length of time, where the time values can be configured within [48] hours;*

  ];
- *IKEv2 SA lifetimes can be configured by a Security Administrator based on*

  [

  - *length of time, where the time values can be configured within [48]hours*

  ]

].

FCS_IPSEC_EXT.1.8

The TSF shall ensure that [selection:

- *IKEv1 Phase 2 SA lifetimes can be configured by a Security Administrator based on* [
  - *length of time, where the time values can be configured within [48]hours;*

  ];
- *IKEv2 Child SA lifetimes can be configured by a Security Administrator based on* [
  - *length of time, where the time values can be configured within [48]hours;*

  ]

].

FCS_IPSEC_EXT.1.9

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

FCS_IPSEC_EXT.1.10

The TSF shall generate nonces used in [*IKEv1, IKEv2*] exchanges of length [

- *according to the security strength associated with the negotiated Diffie-Hellman group;*
- *at least 128 bits in size and at least half the output size of the negotiated pseudorandom function (PRF) hash*

].

FCS_IPSEC_EXT.1.11

The TSF shall ensure that IKE protocols implement DH Group(s) [

- [*14 (2048-bit MODP), 15 (3072-bit MODP)*] according to RFC 3526

].

FCS_IPSEC_EXT.1.12

The TSF shall be able to ensure by default that the strength of the symmetric algorithm (in terms of the number of bits in the key) negotiated to protect the [*IKEv1 Phase 1, IKEv2 IKE_SA*] connection is greater than or equal to the strength of the symmetric algorithm (in terms of the number of bits in the key) negotiated to protect the [*IKEv1 Phase 2, IKEv2 CHILD_SA*] connection.

FCS_IPSEC_EXT.1.13

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

FCS_IPSEC_EXT.1.14

The TSF shall only establish a trusted channel if the presented identifier in the received certificate matches the configured reference identifier, where the presented and reference identifiers are of the following fields and types: [*SAN: IP address, SAN: Fully Qualified Domain Name (FQDN), SAN: user FQDN] and* [*no other reference identifier type*].

5.2.2.9 FCS_RBG_EXT.1 Random Bit Generation

FCS_RBG_EXT.1.1

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

FCS_RBG_EXT.1.2

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

5.2.2.10 FCS_SSHS_EXT.1 SSH Server Protocol

FCS_SSHS_EXT.1.1

The TSF shall implement the SSH protocol in accordance with: RFCs 4251, 4252, 4253, 4254, [*4256, 4344, 6668, 8268, 8308* section 3.1, 8332].

FCS_SSHS_EXT.1.2

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

Application Note: Altered by TD0631

FCS_SSHS_EXT.1.3

The TSF shall ensure that, as described in RFC 4253, packets greater than *[65000]* bytes in an SSH transport connection are dropped.

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: [*aes128-cbc, aes256-cbc, aes128-ctr, aes256-ctr*].

FCS_SSHS_EXT.1.5 –

The TSF shall ensure that the SSH public-key based authentication implementation uses [*ssh-rsa*] as its public key algorithm(s) and rejects all other public key algorithms.

FCS_SSHS_EXT.1.6

The TSF shall ensure that the SSH transport implementation uses [*hmac-sha1, hmac-sha2-256, hmac-sha2-512*] as its MAC algorithm(s) and rejects all other MAC algorithm(s).

FCS_SSHS_EXT.1.7

The TSF shall ensure that [*diffie-hellman-group14-sha1*] and [*diffie-hellman-group14-sha256, diffie-hellman-group16-sha512*] 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 each encryption key is used to protect no more than one gigabyte of data. After any of the thresholds are reached, a rekey needs to be performed.

5.2.3 Identification and Authentication (FIA)

5.2.3.1 FIA_AFL.1 Authentication Failure Management

FIA_AFL.1.1

The TSF shall detect when an Administrator configurable positive integer within *[1-64]* unsuccessful authentication attempts occur related to *Administrators attempting to authenticate remotely using a password*.

FIA_AFL.1.2

When the defined number of unsuccessful authentication attempts has been met, the TSF shall [*prevent the offending Administrator from successfully establishing a remote session using any authentication method that involves a password until an Administrator defined time period has elapsed*].

5.2.3.2 FIA_PMG_EXT.1 Password Management

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 between [*6*] and [*50*] characters.

5.2.3.3 FIA_UIA_EXT.1 User Identification and Authentication

FIA_UIA_EXT.1.1

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

- Display the warning banner in accordance with FTA_TAB.1;
- [*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.

5.2.3.4 FIA_UAU_EXT.2 Password-based Authentication Mechanism

FIA_UAU_EXT.2.1

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

5.2.3.5 FIA_UAU.7.1 Protected Authentication Feedback

FIA_UAU.7.1

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

5.2.3.6 FIA_X509_EXT.1/Rev X.509 Certificate Validation

FIA_X509_EXT.1.1/Rev

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

- RFC 5280 certificate validation and certification path validation **supporting a minimum path length of three certificates** .
- The certification path must terminate with a trusted CA certificate designated as a trust anchor.
- The TSF shall validate a certification path by ensuring that all CA certificates in the certification path contain the basicConstraints extension with the CA flag set to TRUE.

- 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 extendedKeyUsagefield.*
    - *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.

5.2.3.7 FIA_X509_EXT.2 X.509 Certificate Authentication

FIA_X509_EXT.2.1

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

FIA_X509_EXT.2.2

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

5.2.3.8 FIA_X509_EXT.3 X.509 Certificate Requests

FIA_X509_EXT.3.1

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

FIA_X509_EXT.3.2

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

5.2.4 Security Management (FMT)

5.2.4.1 FMT_MOF.1/Functions Management of Security Functions Behaviour

FMT_MOF.1.1/Functions

The TSF shall restrict the ability to [*modify the behaviour of*] the functions [*transmission of audit data to an external IT entity*] to *Security Administrators*.

5.2.4.2 FMT_MOF.1/ManualUpdate Management of Security Functions Behaviour

FMT_MOF.1.1/ManualUpdate

The TSF shall restrict the ability to enable the function *to perform manual updates to Security Administrators*.

5.2.4.3 FMT_MOF.1/Services Management of Security Functions Behaviour

FMT_MOF.1.1/Services

The TSF shall restrict the ability to **start and stop** ~~the functions~~ **services** to *Security Administrators.*

5.2.4.4 FMT_MTD.1/CoreData Management of TSF Data

FMT_MTD.1.1/CoreData

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

5.2.4.5 FMT_MTD.1/CryptoKeys Management of TSF Data

FMT_MTD.1.1/CryptoKeys

The TSF shall restrict the ability to *manage* the *cryptographic keys* to *Security Administrators*.

5.2.4.6 FMT_SMF.1 Specification of Management Functions

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 [hash comparison] capability prior to installing those updates;*
- *Ability to configure the authentication failure parameters for FIA_AFL.1;*
- [
    - *Ability to start and stop services;*
    - *Ability to configure audit behaviour (e.g. changes to storage locations for audit; changes to behaviour when local audit storage space is full);*
    - *Ability to modify the behaviour of the transmission of audit data to an external IT entity;*
    - *Ability to manage the cryptographic keys;*
    - *Ability to configure the cryptographic functionality;*
    - *Ability to configure thresholds for SSH rekeying;*
    - *Ability to configure the lifetime for IPsec SAs;*
    - *Ability to re-enable an Administrator account;*
    - *Ability to set the time which is used for time-stamps;*
    - *Ability to configure the reference identifier for the peer;*
    - *Ability to manage the TOE's trust store and designate X509.v3 certificates as trust anchors;*
    - *Ability to import X.509v3 certificates to the TOE's trust store;*
    - *Ability to manage trusted public keys database;*
    - *No other capabilities*].

Application Note: Altered by TD0631

5.2.4.7 FMT_SMR.2 Restrictions on Security Roles

FMT_SMR.2.1

The TSF shall maintain the roles:

- *Security Administrator*

FMT_SMR.2.2

The TSF shall be able to associate users with roles.

FMT_SMR.2.3

The TSF shall ensure that the conditions

- *The Security Administrator role shall be able to administer the TOE locally;*
- *The Security Administrator role shall be able to administer the TOE remotely;*

are satisfied.

5.2.5 Protection of the TSF (FPT)

5.2.5.1 FTP_APW_EXT.1 Protection of Administrator Passwords

FPT_APW_EXT.1.1

The TSF shall store administrative passwords in non-plaintext form.

FPT_APW_EXT.1.2

The TSF shall prevent the reading of plaintext administrative passwords.

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

FPT_SKP_EXT.1.1

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

5.2.5.3 FPT_STM_EXT.1 Reliable Time Stamps

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*].

5.2.5.4 FPT_TST_EXT.1 TSF Testing

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: [*Data bus test*, *RAM test*, *SRAM test*, *Software integrity check*, *FIPS POST KATS*, and *FIPS POST DRBG*].

5.2.5.5 FPT_TUD_EXT.1 Trusted Update

FPT_TUD_EXT.1.1

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

FPT_TUD_EXT.1.2

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

FPT_TUD_EXT.1.3

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

5.2.6 TOE Access (FTA)

5.2.6.1 FTA_SSL_EXT.1 TSF-initiated Session Locking

FTA_SSL_EXT.1.1

The TSF Shall, for local interactive sessions, [

- *terminate the session*]

after a Security Administrator-specified time period of inactivity

5.2.6.2 FTA_SSL.3 TSF-initiated Termination

FTA_SSL.3.1

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

5.2.6.3 FTA_SSL.4 User-initiated Termination

FTA_SSL.4.1

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

5.2.6.4 FTA_TAB.1 Default TOE Access Banners

FTA_TAB.1.1

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

5.2.7 Trusted Path/Channels (FTP)

5.2.7.1 FTP_ITC.1 Inter-TSF Trusted Channel

FTP_ITC.1.1

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

FTP_ITC.1.2

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

FTP_ITC.1.3

The TSF shall initiate communication via the trusted channel for *[logging service on the remote audit server, authentication service]*.

5.2.7.2 FTP_TRP.1/Admin Trusted Path

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*.

5.3 TOE SFR Dependencies Rationale for SFRs

The PP contain all the requirements claimed in this ST. As such, the dependencies are not applicable since the PP has been approved.

5.4 Security Assurance Requirements

The TOE assurance requirements for this ST are taken directly from the PP, which are derived from Common Criteria Version 3.1, Revision 5. The assurance requirements are summarized in Table 14.

Table 14 – Security Assurance Requirements

Assurance Class	Assurance Components	Component Description
Security Target	ASE_CCL.1	Conformance claims
	ASE_ECD.1	Extended components definition
	ASE_INT.1	ST introduction
	ASE_OBJ.1	Security objectives for the operational environment
	ASE_REQ.1	Stated security requirements
	ASE_SPD.1	Security problem definition
	ASE_TSS.1	TOE Summary Specification
Development	ADV_FSP.1	Basic functionality specification

Assurance Class	Assurance Components	Component Description
Guidance Documents	AGD_OPE.1	Operational user guidance
	AGD_PRE.1	Preparative Procedures
Life Cycle Support	ALC_CMC.1	Labelling of the TOE
	ALC_CMS.1	TOE CM coverage
Tests	ATE_IND.1	Independent testing – conformance
Vulnerability Assessment	AVA_VAN.1	Vulnerability survey

5.5 Assurance Measures

The TOE satisfied the identified assurance requirements. This section identifies the Assurance Measures applied by Nokia Corporation to satisfy the assurance requirements. The following table lists the details.

Table 15 – TOE Security Assurance Measures

SAR Component	How the SAR will be met
ADV_FSP.1	The functional specification describes the external interfaces of the TOE; such as the means for a user to invoke a service and the corresponding response of those services. The description includes the interface(s) that enforces a security functional requirement, the interface(s) that supports the enforcement of a security functional requirement, and the interface(s) that does not enforce any security functional requirements. The interfaces are described in terms of their purpose (general goal of the interface), method of use (how the interface is to be used), parameters (explicit inputs to and outputs from an interface that control the behavior of that interface), parameter descriptions (tells what the parameter is in some meaningful way), and error messages (identifies the condition that generated it, what the message is, and the meaning of any error codes).
AGD_OPE.1	The Administrative Guide provides the descriptions of the processes and procedures of how the administrative users of the TOE can securely administer the TOE using the interfaces that provide the features and functions detailed in the guidance.
AGD_PRE.1	The Installation Guide describes the installation, generation, and startup procedures so that the users of the TOE can put the components of the TOE in the evaluated configuration.
ALC_CMC.1 ALC_CMS.1	The Configuration Management (CM) documents describe how the consumer identifies the evaluated TOE. The CM documents identify the configuration items, how those configuration items are uniquely identified, and the adequacy of the procedures that are used to control and track changes that are made to the TOE. This includes details on what changes are tracked and how potential changes are incorporated.
ATE_IND.1	Nokia will provide the TOE for testing.
AVA_VAN.1	Nokia will provide the TOE for testing. Nokia will provide a document identifying the list of software and hardware components.

6 TOE Summary Specification

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

Table 16 – TOE Summary Specification SFR Description

Requirement	TSS Description
FAU_GEN.1	The TOE produces audit events for start-up and shutdown of the audit functions as well as the following: administrative login and logout; password resets; changes to the TOE data related to configuration; the generation, import of, changing, or deletion of cryptographic keys. Audit records include the identity of the administrator initiating the cryptography related events such as, key generation (e.g. RSA), import, or deletion. The audit record contains the information such as, the identity of the key (unique name including the size and type), the date and time of the event, type of event, and the outcome of the event. Following is an example of an audit record for key generation: 189 2021/03/24 16:26:41.961 UTC MINOR: SECURITY #2231 management admin “admin certificate gen-keypair cf3:/key_1 size 2048 type rsa : success” Following is an example of an audit record for key import: 197 2021/03/24 17:35:22.606 UTC MINOR: SECURITY #2232 management admin "admin certificate import type key input cf3:/key_1.pem output key_1.pem format pem : success" Following is an example of an audit record for key deletion: 198 2021/03/24 17:36:53.864 UTC MINOR: SECURITY #2234 management admin "File cf3-A:\system-pki\key_1.pem delete : success". Only Authorized Administrators can access the audit events and have the ability to clear the audit events. The TOE creates audit records for events and provides contents as required for all SFRs specified in Table 13.
FAU_GEN.2	For audit events that result from actions of identified users, the TOE can associate each auditable event with the identity of the user that caused the event.
FAU_STG_EXT.1	The TOE is a standalone TOE that is configured to export audit data to a specified, external audit server. The TOE protects communications with an external audit server via IPSEC. The TOE maintains a circular buffer of a maximum of 3000 audit records which is exported to a syslog server in real time. If for any reason the number of entries exceeds the capacity (i.e. the maximum number of audit records), the circularity of the buffer ensures that the oldest entries are overwritten when new entries are generated and stored. Only authorized

<table>
<tr><th>Requirement</th><th>TSS Description</th></tr>
<tr><td></td><td>Administrators may access the audit records, unprivileged users have no access rights to the log files.</td></tr>
<tr><td>FCS_CKM.1</td><td>To support the cryptographic protocols, the TOE use:
1. RSA schemes using cryptographic key sizes of 2048-bit that meet the FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Appendix B.3; and
2. FFC Schemes using ‘safe-prime’ groups that meet the following: “NIST Special Publication 800-56A Revision 3 and RFC 3526”.
For SSH communications, the RSA keys are used in support of digital signatures.
For IPSec communications, the RSA keys are used when authenticating remote peers.
The FFC safe-prime groups are used in both SSH and IPSec.
The relevant NIST CAVP certificate numbers are listed in Table 5.</td></tr>
<tr><td>FCS_CKM.2</td><td>The TOE performs cryptographic key establishment in accordance with RSA key establishment schemes that are conformant to the following:
1. RSA-based key establishment schemes that meet the following: RSAES-PKCS1-v1_5 as specified in Section 7.2 of RFC 3447, “Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1”; and
2. FFC Schemes using “safe-prime” groups that meet the following: ‘NIST Special Publication 800-56A Revision 3, “Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography” and [RFC 3526].
<table>
<tr><td>Scheme</td><td>SFR</td><td>Services</td></tr>
<tr><td>FFC-Safe Primes /DHG14
FFC-Safe Primes /DHG16</td><td>FCS_SSHS_EXT.1</td><td>Administration</td></tr>
<tr><td>FFC-Safe Primes /DHG14
FFC-Safe Primes /DHG15</td><td>FCS_IPSEC_EXT.1</td><td>Administration</td></tr>
<tr><td rowspan="2">RSA</td><td>FCS_SSHS_EXT.1</td><td>Administration</td></tr>
<tr><td>FCS_IPSEC_EXT.1</td><td>Audit server, Authentication server</td></tr>
</table>
For additional details on CAVP Certificate mapping, refer to Table 5.</td></tr>
<tr><td>FCS_CKM.4</td><td>The TOE destroys all cryptographic keys using the following methods:
• For plaintext keys in volatile storage, the TOE uses a single overwrite consisting of zeroes.
• For all plaintext keys in non-volatile storage, the TOE destroys keys via invocation of an interface provided by a part of the TOE that instructs TOE to destroy the abstraction that represents the key.
Please refer to Table 17 for key zeroization.</td></tr>
</table>

<table>
<tr><th>Requirement</th><th>TSS Description</th></tr>
<tr><td>FCS_COP.1/DataEncryption</td><td>The TOE supports AES encryption and decryption conforming to ISO 18033-3, ISO 10116 and ISO 19772.
The TOE provides AES encryption and decryption in support of SSHv2 for secure communications. The AES key sizes supported for SSH are 128 bits and 256 bits and the AES modes supported are: CBC and CTR.
In addition, AES_CBC is used in as a Cryptographic Algorithm selection for ESP protocol. The key sizes are 128-bits, 192-bits, and 256-bits.
The relevant NIST CAVP certificate numbers are listed in Table 5.</td></tr>
<tr><td>FCS_COP.1/Hash</td><td>The TOE supports Cryptographic hashing services conforming to ISO/IEC 10118-3:2004. The hashing algorithms are used in SSH connections for secure communications and for IPSec.
The following hashing algorithms are supported: SHA-1, SHA-256, SHA-384, and SHA-512.
The message digest sizes supported are: 160, 256, 384, and 512 bits.
The relevant NIST CAVP certificate numbers are listed in Table 3.</td></tr>
<tr><td>FCS_COP.1/KeyedHash</td><td>The TOE performs keyed-hash message authentication in accordance with ISO/IEC 9797-2:2011, Section 7 “MAC Algorithm 2”.
The details of Key Size and Message Digest Size are given below with the respective HMAC Algorithm.
<table>
<tr><td>HMAC Algorithm</td><td>Hash Function</td><td>Block Size</td><td>Key Lengths</td><td>MAC Lengths</td></tr>
<tr><td>HMAC-SHA-1</td><td>SHA-1</td><td>512 bits</td><td>160 bits</td><td>160 bits</td></tr>
<tr><td>HMAC-SHA-256</td><td>SHA-256</td><td>512 bits</td><td>256 bits</td><td>256 bits</td></tr>
<tr><td>HMAC-SHA-384</td><td>SHA-384</td><td>1024 bits</td><td>384 bits</td><td>384 bits</td></tr>
<tr><td>HMAC-SHA-512</td><td>SHA-512</td><td>1024 bits</td><td>512 bits</td><td>512 bits</td></tr>
</table>
The TOE leverages HMAC algorithm in support of IPSEC and SSH sessions.
The relevant NIST CAVP certificate numbers are listed in Table 5.</td></tr>
<tr><td>FCS_COP.1/SigGen</td><td>The TOE supports cryptographic signature services such as generation and verification using RSA Digital Signature Algorithm that meet the RSA scheme specified in FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Section 5.5, using PKCS #1 v2.1 Signature Schemes RSASSA-PKCS1v1_5.
The RSA key size supported is 2048 bits. RSA signature generation is used with SSH Public Key Authentication.
The relevant NIST CAVP certificate numbers are listed in Table 5.</td></tr>
<tr><td>FCS_IPSEC_EXT.1</td><td>The TOE implements IPSec in accordance with RFC 4301 in tunnel mode only.
Incoming packets are inspected against the session database. Sessions that match all the security attributes and do not exceed the TTL are automatically passed on to their destination and are send via a VPN</td></tr>
</table>

Requirement	TSS Description
	interface where applicable. Packets that do not match the attributes in the session database are then compared to the configured Access Control Lists for that interface identifier and the direction (ingress or egress) based on the ACL entry ID's numerical value. Packets that are permitted are passed to their destination, packets that matches ACLs marked for logging are written to the audit log and packets marked for dropping are discarded. The TOE permits two actions to be assigned to Access Control Lists – Permit (allow the packet to flow through the TOE with no protection) and Drop (drop the packet with no further processing). The ACLs can be applied to the IPSec VPN tunnels. The TOE implements AES-CBC-128, AES-CBC-192 and AES-CBC-256 using HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384 and HMAC-SHA-256 for ESP protection. Both IKEv1 and IKEv2 are implemented. IKEv1 as defined in RFCs 2407, 2408, 2409, RFC 4109 and RFC 4868 for hash functions; IKEv2 as defined in RFCs 5996 (with support for NAT traversal) and RFC 4868 for hash functions. IKEv1 aggressive mode is not supported, and only main mode is permitted in the evaluated configuration. The TOE implements AES-CBC-128, AES-CBC-192 and AES-CBC-256 for payload protection in IKEv1 and IKEv2. In the evaluated configuration, the TOE permits configuration of the: • IKEv1 Phase 1 and IKEv2 SA lifetimes in terms of length of time (1,200 to 172,800 seconds), • IKEv1 Phase 2 SA and IKEv2 Child SA lifetimes in terms of length of time (1,200 to 172,800 seconds) The lifetime of the IPsec SA is configured through the CLI only accessible to the administrators of the TOE. The TOE utilizes CTR-DRBG with AES (as specified in FCS_RBG_EXT.1) to generate the exponents used in IKE key exchanges, having the possible lengths of 224 or 256 bits, corresponding to each of the supported DH groups. For Diffie-Hellman Groups 14 and 15 the TOE always generates value x of 256 bits. For DH Group 14 only 224 of those bits are used. For DH Group 15 all 256 bits are used. The TOE uses HMAC DRBG with SHA-1, SHA-256, SHA-384 and SHA-512 for the generation of DH exponents and nonces. Nonces in the IKE key exchange protocol are always of length 256 bits, whether for DH Group 14 or DH Group 15. The generation of random bits is described at FCS_RBG_EXT.1. The TOE implements Diffie-Hellman Groups 14 and 15. In the IKEv1 phase 1 and phase 2 exchanges, the TOE and peer will agree on the best DH group both can support. When the TOE receives an IKE proposal, it will select the first DH group that matches the acceptable DH groups. The negotiation will fail if there is no match. Similarly, when the peer initiates the IKE protocol, the TOE will select the first match from the IKE proposal sent by the peer and the negotiation fails is no acceptable match is found. The TOE checks the strengths of the configured IKE algorithms prior to committing a tunnel configuration. This ensures that the strength of the symmetric algorithm (128, 192 or 256 bits) negotiated to protect the IKEv1

Requirement	TSS Description
	Phase 1 or IKEv2 IKE_SA connection is greater than or equal to the strength of the symmetric algorithm negotiated to protect the IKEv1 Phase 2 or IKEv2 CHILD_SA connection. If the strength is not greater, an error is displayed, and the configuration fails. The TOE permits peer authentication via RSA that use X.509v3 certificates that conform to RFC 4945 or pre-shared keys. The TOE permits peer authentication via X.509v3 certificates with RSA are used as defined in RFC 5280. Certificate Request Messages are generated in accordance with RFC 2986 when validating certificates for IPsec connections. The use of certificates is described in FIA_X509_EXT.1/Rev and FIA_X509_EXT.3. The TOE accepts bit-based pre-shared keys. The TOE converts text-based pre-shared keys into an authentication value as per RFC 2409 for IKEv1 or RFC 4306 for IKEv2, using SHA-1 or the PRF that is configured as the hash algorithm for the IKE exchanges. When using certificates for peer authentication, the TOE will only establish a trusted channel to peers that provide a valid certificate. The TOE will compare the reference identifier of the peer against the reference identifier stored in the associated certificate. If the two values are not a match, the TOE will not establish the connection. The TOE supports SAN:IP addresses, SAN:FQDNs, and SAN:User FQDN (email address) reference identifiers. The relevant NIST CAVP certificate numbers are listed in Table 5.
FCS_RBG_EXT.1	The TOE produces all deterministic random bit generation services in accordance with ISO/IEC 18031:2011 using CTR_DRBG (AES). The TOE uses deterministic RBG, which is seeded by two entropy sources that accumulate entropy. The sources of entropy are from a software-based noise source and a hardware-noise sources. The CTR_DRBG seeded with a minimum of 256 bits of entropy.
FCS_SSHS_EXT.1	The TOE implements SSH protocol that complies with RFC(s) 4251, 4252, 4253, 4254, 4256, 4344, 6668, 8268, 8308 (Section 3.1) and 8332. The TOE implements public key authentication (SSH-RSA) and password-based authentication. The TOE ensures that packets greater than 65000 bytes in an SSH transport connection are dropped as described in RFC 4253. The TOE supports the following encryption algorithms: aes128-cbc, aes256-cbc aes128-ctr, and aes256-ctr for SSH transport. The TOE supports the following data integrity MAC algorithms: hmac-sha1, hmac-sha2-256, hmac-sha2-512. The TOE supports diffie-hellman-group-14-sha1, diffie-hellman-group-14-sha256 and diffie-hellman-group-16-sha512 for SSH key exchanges. The following public key algorithms are supported: ssh-rsa. The TOE is capable of rekeying. The TOE verifies the following thresholds: • No longer than one hour

Requirement	TSS Description
	• No more than one gigabyte of transmitted data The TOE continuously checks both conditions. When either of the conditions are met, the TOE will initiate a rekey. The SSH client’s public key is compared to an authorized keys file which is stored on the TOE when establishing a user identity.
FIA_AFL.1	When user authentication fails consecutively, the TOE locks the claimed account until the configured time has elapsed Administrators can configure the maximum number of consecutive failed authentication attempts to be between 1 and 64. Administrators may also configure the time period until the counter is reset in case of no further authentication attempts are made. The time may be configured between 0 to 60 minutes. When the account is locked, the TOE does not permit any further actions until the account is accessible. The lockout time may be configured between 0 to 1440 minutes. The authentication failures cannot lead to a situation where no administrator access is available. A user would be configured to still be granted local access to the TOE even if the remote access is denied.
FIA_PMG_EXT.1	The TOE provides the following password management capabilities for administrator passwords: a) Passwords can be composed of any combination of upper and lower case letters, numbers, and the following special characters: “!”, “@”, “#”, “$”, “%”, “^”, “&”, “*”, “(“, “)”. b) Minimum password length configurable to between 6 to 50 characters.
FIA_UIA_EXT.1 FIA_UAU_EXT.2	The TOE only allows limited actions prior to the user being successfully authenticated as an Administrator. The actions allowed on behalf of an unauthenticated used include viewing of the access banner at the login prompt by both local and remote users. Local users are additionally allowed to perform static route configuration, IP address configuration, FIPS Mode configuration, Auto discover/negotiation for interfaces, DNS, Duplex vs. Multiplex, and Image path configuration The TOE mandates that every user must be authenticated by accessing the local console or by remotely using SSH. Security Administrators can access the console by connecting to the console port using RJ45-DB9 or by remotely connecting to each appliance via SSHv2 Users are required to enter a username and password when remotely and locally logging into the TOE. The TOE implements RSA public key authentication via SSHv2, and password-based authentication for remote and local authentication. The user can access the TOE with correct public key-based authentication. For the password-based authentication, users must provide the correct credentials before accessing the TOE. If the user enters incorrect user credentials, they will not be allowed to access and will be presented the login page again.

Requirement	TSS Description
FIA_UAU.7	When a user enters their password, the information is obscured. For remote session authentication, the TOE does not echo any characters when they are entered.
FIA_X509_EXT.1/Rev	The TOE supports the X.509v3 certificates as defined by RFC 5280 to support authentication of external IPSEC peers. When an X.509 certificate is presented, the TOE verifies the certificate path, and certification validation process by verifying the following rules: • RFC 5280 certificate validation and certification path validation supporting a minimum path length of three certificates. • The certification path must terminate with a trusted CA certificate designated as a trust anchor. • The TOE validates a certification path by ensuring that all CA certificates in the certification path contain the basicConstraints extension with the CA flag set to TRUE. • The TOE validates the revocation status of the certificate using a Certificate Revocation List (CRL) as specified in RFC 5280 Section 6.3. • The TOE validates the extendedKeyUsage field according to the following rules: o Server certificates presented for TLS must have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field. o Client certificates presented for TLS must have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the extendedKeyUsage field. o Certificates used for trusted updates and executable code integrity verification shall have the Code Signing purpose (id-kp 3 with OID 1.3.6.1.5.5.7.3.3) in the extendedKeyUsage field. The TOE will only treat a certificate as a CA certificate if the basicConstraints extension is present and the CA flag is set to TRUE. When the TOE receives a remote certificate during the secure channel establishment, the validity of the remote entity certificate is verified. The TOE also verifies the chain of trust by validating each certificate contained in the chain and verifying that a certificate path consists of trusted CA certificates and verify the validity of the certificates. These checks are done prior to loading the certificates onto the TOE. The use of CRL is configurable and can be used for certificate revocation.
FIA_X509_EXT.2	X.509 certificates are used to validate the identities of the communicating peers for the establishment of an IPsec connection. RSA based certificates are supported for the key length of 2048 bits. To validate a peer certificate when CRLs are used, the Security Administrator imports its CA certificates and CRLs. CRL is used to determine whether the certificate is revoked or not. If the CRL cannot be obtained, then the TOE will reject the certificate. CRLs are obtained from a

Requirement	TSS Description
	web server over HTTP and are refreshed according to the CRL update-interval set in the TOE CLI. Revocation check is performed on end-entity and intermediate certificates. If the TOE is unable to establish a connection to determine the validity of a certificate, the TOE will not accept the certificate. CA profiles must be created and enabled for each imported CA certificates (Root CAs and the intermediate CAs) and CRL. The Security Administrator must configure at least one trust anchor to limit the list of CA certificates. Furthermore, the Security Administrator can create a client profile to specify the cipher-list and client certificate to use. If the TOE is unable to establish a connection to determine the validity of a certificate, the TOE will not accept the certificate.
FIA_X509_EXT.3	The TOE generates a Certificate Request as specified by RFC 2986 and is be able to provide the following information in the request: public key and Common Name, Organization, Organizational Unit,Country and State. The TOE validates the chain of certificates from the Root CA upon receiving the CA Certificate Response. The TOE does not support the “device-specific information” within Certificate Request message.
FMT_MOF.1/Functions	The TOE restricts the ability to modify the behavior of transmission of audit data to an audit server to Security Administrators.
FMT_MOF.1/ManualUpdate	Security Administrators can perform manual software updates to the product. This is performed via the command line. When triggered, the product downloads updates from a configured update server, performs an integrity test, and installs the new software.
FMT_MOF.1/Services	The Security Administrator may use the CLI to start and stop the services. The following services may be started and stopped: • SSH server • SFTP server • SNMP • Logging to file, memory, syslog • IPSec tunnels • RADIUS • TACACS+
FMT_MTD.1/CoreData	The TOE implements Role Based Access Control (RBAC). Only the role Security Administrator is implemented. When a user with sufficient credentials is successfully authenticated, the user shall enter the role Security Administrator. Only Security Administrators can manage the certificates in TOE’s trust store. Security Administrator role is associated with a set of defined access rights that are granted to a user entering the role. The access rights grant the user a right to access TOE data and functions. The TOE prevents users not assigned to the Security Administrator role from accessing the TOE data and functions requiring Security Administrator access rights.

Requirement	TSS Description
	Accesses available to unauthenticated users depend on the access methods. When accessing the TOE remotely, the unauthenticated user is only allowed to view the TOE access banner and login prompts. For the users accessing the TOE locally, the TOE allows viewing of the access banner and the login prompts but also allows the user to access basic configuration tasks of the TOE, including Static route configuration, IP address configuration, FIPS Mode configuration, Auto discover/negotiation for interfaces, DNS, Duplex vs. Multiplex, and Image path configuration. The TOE implements active and inactive trust stores. Trust stores In the volatile memory are active. Inactive trust stores reside in the persistent store in the form of files.
FMT_MTD.1/CryptoKeys	The Security Administrator has the ability to modify, generate, and delete SSH and IPsec session keys as well as any configured X.509 certificates. The key management functions are performed through the Command Line Interface and not accessible to users in other roles.
FMT_SMF.1	The TOE only implements the role Security Administrator. The TOE can be accessed by users assigned to the role Security Administrator through a Command Line Interface locally from console or remotely over SSH. The Administrator can perform 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 hash comparison prior to installing those updates; • Ability to configure the authentication failure parameters for FIA_AFL.1; • Ability to start and stop services; • Ability to configure audit behaviour (e.g. changes to storage locations for audit; changes to behaviour when local audit storage space is full); • Ability to modify the behaviour of the transmission of audit data to an external IT entity; • Ability to manage the cryptographic keys; • Ability to configure the cryptographic functionality; • Ability to configure thresholds for SSH rekeying; • Ability to configure the lifetime for IPsec SAs; • Ability to re-enable an Administrator account; • Ability to set the time which is used for time-stamps; • Ability to configure the reference identifier for the peer; • Ability to manage the TOE's trust store and designate X509.v3 certificates as trust anchors; and • Ability to import X.509v3 certificates to the TOE's trust store.
FMT_SMR.2	The TOE maintains a single role, Security Administrator. Each user is identified with a username and authenticated with a password prior to being

Requirement	TSS Description
	assigned to a role. Only successfully authenticated users shall be assigned to roles. User assigned to the role Security Administrator may administer the TOE locally and remotely. Local access is via console, remote access is over SSH. Admins can configure user's privilege that grant or deny privilege to users from login access via Console and Remote access.
FPT_APW_EXT.1	The passwords are stored in files in non-reversible hashes. This ensures that they cannot be recovered from the files. When a password is read from the CLI, it is hashed and the hash is compared to the reference value stored in the password file. This ensures that the passwords are not stored in plaintext.
FPT_SKP_EXT.1	The TOE stores all private keys in a secure storage and is not accessible through an interface to administrators.
FPT_STM_EXT.1	The TOE provides reliable time stamps. The clock function is reliant on the system clock provided by the underlying hardware. The clock is utilized for providing reliable time stamps used in the following functions: • Audit events • Session inactivity • X.509 certificate expiration validation.
FPT_TST_EXT.1	The TOE executes the integrity check of the installed firmware by comparing the published HMAC-SHA256. If the hash does not match, the inactive CPM will reboot continuously until the CF is replaced with an authentic firmware. The TOE also performs self-tests for the cryptographic module during boot up, and if any component reports failure for the self-test, the system will reboot and display the appropriate information on the local console. All ports are blocked from moving to forwarding state during the POST. If all components of all modules pass the POST, the system is placed in FIPS PASS state and ports are allowed to forward data traffic. When any of the tests fail, a message is displayed to the local console. The TOE executes the following power-on self-tests: • Software integrity test: For this test, when the CPM boots up, the bootloader calculates the HMAC-SHA256 authentication code of that software image from the storage and compares it with the known value stored in storage. If the value is not the same then it will give an error which is present on the console, and then the device will reboot. If the values of HMAC-SHA256 matches, then it successfully executes the software image. • AES Known Answer Test -The AES encryption and AES decryption algorithms are tested using test vectors. The results are compared against pre-computed results to ensure the algorithms are operating properly. • CMAC Known Answer Test - With this test, the CMAC authentication code is generated for a known message and respected key. Both are compared to the expected authentication code, if they match the test

Requirement	TSS Description
	gets passed and if they do not the test get failed. The message is displayed on the console screen. • GCM Known Answer Test - In this test, A known plain-text is encrypted using AES-GCM with a known 256-bit key, and the computed cipher-text is compared to the expected cipher-text. If they match, then the computed cipher-text is decrypted using the same key, and the recovered plaintext is compared with the original known plain-text. If they do not match, the test fails. If they match, the test passes. • CCM Known Answer Test - In this test, the known plain text is encrypted using the AES-CCM with known 192 bits key, and then the computed cipher text is compared against the expected cipher txt . If they match, then the computed cipher-text is decrypted using the same key, and the recovered plaintext is compared with the original known plain-text. If they do not match, the test fails. If they match, the test passes. • HMAC-SHA-1/224/256/384/512 Known Answer Test - the HMAC algorithm is tested using test vector. The results are compared against pre-computed results to ensure the algorithm is operating properly. • SHA-1/256/512 Known Answer Test - the SHA algorithm is tested using test vector. The results are compared against pre-computed results to ensure the algorithm is operating correctly. • RSA Signature Known Answer Test - the RSA Signature is tested using test vector. The results are compared against pre-computed results to ensure the algorithm is operating properly. • DRBG Known Answer Test - the DRBG is seeded with a pre-determined entropy and the RNG output is compared with output values expected for the pre-determined seed. is also executed as part of self-tests. • The Software Integrity Test - is run automatically on start-up, and whenever the system images are loaded. These tests are sufficient to verify that the correct version of the TOE software is running as well as that the cryptographic operations are all performing as expected. • There is also a Noise Source Health test that is executed as part of the self-test requirements.
FPT_TUD_EXT.1	The TOE implements a "show version" CLI command for the Security Administrators to query the current version of the TOE software. The administrators may also perform manual software updates. The TOE does not implement functions for automatically upgrading the software, each upgrade must be performed manually by the Administrator. To upgrade the TOE software, the Administrator first connects to the update server using SFTP and downloads the firmware upgrade to a Compact Flash (CF) device. The upgrade is protected by a HMAC-SHA-256 value which is computed by the developer in the development environment and stored in a separate file. The TOE uses a Boot Options File (BOF) for indicating to the boot loader the location of the TOE software. Typically, the Administrator stores the firmware upgrade on a CF and modifies the BOF to point to the software on the CF. This does not need to be performed immediately after

Requirement	TSS Description
	downloading the software upgrade. The Administrator may time the actual upgrading at a convenient time. When rebooting the TOE with the modified BOF, the TOE upgrades the software from the source pointed to by the BOF. When in the FIPS mode, the boot loader searches for the hash file containing the HMAC-SHA-256 value of the TOE software in the same location as the software image. When the HMAC file is found, the TOE computes a HMAC value of the image and compares it to the value on the HMAC file. If the values match, the TOE continues with the boot up. If the HMAC file is not found or the comparison fails, the boot loader reboots the system.
FTA_SSL.3 FTA_SSL_EXT.1	The TOE will terminate a remote interactive session after a configurable time interval of session inactivity. A configured inactivity period will be applied to both local and remote sessions in the same procedure. When the interface has been idle for more than the configured period, the session will be terminated and will require authentication to establish a new session. If a local user session is not active for a configured period of time, the session will be terminated and will require re-identification and authentication to establish a new session. If a remote user session is inactive for a configured period of time, the session will be terminated and will require re-identification and authentication to establish a new session. When the user logs back in, the inactivity timer will be activated for the new session. A configured inactivity period will be applied to both local and remote sessions in the same manner. The allowable inactivity timeout range is from 1 to 1440 minutes.
FTA_SSL.4	The Security Administrator is able to manually terminate their CLI using the command ‘logout’.
FTA_TAB.1	Security Administrators can create a customized login banner that will be displayed at the following interfaces: • Local CLI • Remote CLI This banner will be displayed prior to allowing Security Administrator access through those interfaces.
FTP_ITC.1	The TOE supports secure communication to the following IT entities: Audit server and Authentication server (TACACS+ and RADIUS). The TOE provides secure communication by using IPSEC between itself and Audit server, and between itself and Authentication servers. The TOE uses IPSEC protocol with X.509 certificate-based authentication. The protocols listed are consistent with those specified in the requirement.
FTP_TRP.1/Admin	The TOE supports SSH v2.0 for secure remote administration of the TOE. SSH v2.0 session is encrypted using AES encryption to protect confidentiality and uses HMACs to protect integrity of traffic. The protocols listed are consistent with those specified in the requirement.

6.1 CAVP Details

The CAVP certificate numbers for the cryptographic algorithms the TOE implements are given in Table 5.

6.2 Cryptographic Key Destruction

The table below describes the key zeroization provided by the TOE and as referenced in FCS_CKM.4.

Table 17 – Key Destruction Methods

Keys/CSPs	Purpose	Storage Location	Method of Zeroization
SSH Private host key	SSH Server host private key stored in the local file system	Volatile memory when used, file system for persistent storage	Single overwrite with zeros in volatile memory, file erasure in persistent storage
SSH Session key	Session key loaded into memory to complete a SSH session establishment	Volatile memory	Single overwrite with zeros
RNG state	Internal state and seed key of the DRBG	Volatile memory	Handled by kernel, overwritten with zeros at the boot-up
IKE Private host key	Private authentication used by IKE	In volatile memory when used, in file system when persistently stored	Single overwrite with zeros in volatile memory, not erased when persistently stored
IKE-SKEYID	IKE master secret for deriving IKE and IPsec ESP Session keys	Volatile memory	Erased when a tunnel goes down or zeroized at reboot
IKE Session key	IKE Session keys	Volatile memory	Erased when a tunnel goes down or zeroized at reboot
ESP Session key	ESP Session keys	Volatile memory	Erased when a tunnel goes down or zeroized at reboot
IKE-DH Private exponent	Ephemeral DH private exponent used in IKE	Volatile memory	Erased when a tunnel goes down or zeroized at reboot
SW Integrity key	The key used in the HMAC for verifying the TOE Software integrity.	Shipped with the software image package, resides in the same directory with the software image files. The location is pointed to by the BOF (boot options file) and may be Compact Flash (CF) or an FTP location outside of the TOE.	Replaced with a new value when a new software image package is loaded.

7 Acronym Table

Table 18 – Acronyms

Acronym	Definition
AAA	Authentication, Authorization and Accounting
AES	Advanced Encryption Standard
BOF	Boot Options File
CA	Certification Authority
CBC	Cipher Block Chaining
CC	Common Criteria
CF	Compact Flash
CLI	Command Line Interface
CMAC	Cipher MAC
cPP	collaborative PP
CPU	Central Processing Unit
CRL	Certificate Revocation List
CTR	Counter Mode
DH	Diffie-Hellman
RDBG	Deterministic Random Bit Generator
EP	Extension Package
ESP	Encapsulating Security Payload
FIPS	Federal Information Processing Standard
FD	Flexible Data-Rate
FTP	File Transfer Protocol
gRPC	gRPC Remote Procedure Calls
Gb	Giga-bit
GCM	Galois Counter Mode
HMAC	Hash Message Authentication Code
HTTP	Hyper Text Transfer Protocol
IKE	Internet Key Exchange
IP	Internet Protocol
IPsec	IP Security
KAT	Known Answer Test
MAC	Message Authentication Code
MACsec	MAC Security
Mb	Mega-bit
MPLS	Multiprotocol Layer Switching
NDcPP	Network Device cPP

Acronym	Definition
NIAP	National Information Assurance Partnership
NTP	Network Time Protocol
OCSP	Online Certificate Status Protocol
OSP	Organizational Security Policy
PIN	Personal Identification Number
PKCS	Public Key Cryptography Standard
PKI	Public Key Infrastructure
PoE	Power Over Ethernet
PoE+	PoE Plus
PP	Protection Profile
RADIUS	Remote Authentication Dial-In User Service
RAM	Random Access Memory
RFC	Request For Comments
RSA	Rivest Shamir Adleman
SAR	Security Assurance Requirement or Service Aggregation Router
SFP	Small Form-Factor Pluggable
SFP+	SFP Plus
SFR	Security Functional Requirements
SFTP	Secure File Transfer Protocol
SHA	Secure Hash Algorithm
SNMP	Simple Network Management Protocol
SRAM	Static RAM
SAR OS	Service Router Operating System
SSH	Secure Shell
TACACS+	Terminal Access Controller Access-Control System Plus
TCP	Transport Control Protocol
TD	Technical Decision
TOE	Target of Evaluation
TRNG	True Random Number Generator
TSF	TOE Security Function
TSS	TOE Summary Specification