Nokia 7x50 SR OS 20.10.R12 for 7750 SR-7, 7750 SR-12, 7750 SR-12e, 7750 SR-1e, 7750 SR-2e, 7750 SR-3e, 7750 SR-a4, and 7750 SR-a8 with maxp10-10/1Gb-msec-sfp+ and me12-10/1gb-sfp+ MDAs Security Target

Document Version: 3.4
Date: May 30, 2023

intertek acumen security

2400 Research Blvd
Suite 395
Rockville, MD 20850

Contents

1 Introduction

This document provides the basis for an evaluation of a specific Target of Evaluation (TOE), the Nokia 7x50 SR OS 20.10.R12 for 7750 SR-7, 7750 SR-12, 7750 SR-12e, 7750 SR-1e, 7750 SR-2e, 7750 SR-3e, 7750 SR-a4, and 7750 SR-a8 with maxp10-10/1Gb-msec-sfp+ and me12-10/1gb-sfp+ MDAs. This Security Target (ST) defines a set of assumptions about the aspects of the TOE environment, a list of threats that the TOE intends to counter, a set of security objectives, a set of security requirements, and the IT security functions provided by the TOE that meet that set of requirements. Administrators of the TOE will be referred to as Security Administrators in this document.

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 7x50 SR OS 20.10.R12 for 7750 SR-7, 7750 SR-12, 7750 SR-12e, 7750 SR-1e, 7750 SR-2e, 7750 SR-3e, 7750 SR-a4, and 7750 SR-a8 with maxp10-10/1Gb-msec-sfp+ and me12-10/1gb-sfp+ MDAs Security Target
ST Version	3.4
ST Date	May 30, 2023
ST Author	Acumen Security, LLC.
TOE Identifier	Nokia 7x50 SR OS 20.10.R12 for 7750 SR-7, 7750 SR-12, 7750 SR-12e, 7750 SR-1e, 7750 SR-2e, 7750 SR-3e, 7750 SR-a4, and 7750 SR-a8 with maxp10-10/1Gb-msec-sfp+ and me12-10/1gb-sfp+ MDAs
TOE Hardware	7750 SR-7, 7750 SR-12, 7750 SR-12e, 7750 SR-1e, 7750 SR-2e, 7750 SR-3e, 7750 SR-a4, and 7750 SR-a8
TOE Software	Nokia SR OS 20.10.R12
TOE Developer	Nokia Corporation
Key Words	Network Device, Nokia, Encryption, SR OS, MACsec

1.2 TOE Overview

The Nokia 7x50 SR OS 20.10.R12 for 7750 SR-7, 7750 SR-12, 7750 SR-12e, 7750 SR-1e, 7750 SR-2e, 7750 SR-3e, 7750 SR-a4, and 7750 SR-a8 with maxp10-10/1Gb-msec-sfp+ and me12-10/1gb-sfp+ MDAs (herein referred to as the TOE) is a network device with the high-performance, scale and flexibility supporting service providers, web scale and enterprise networks. The Nokia 7x50 routers utilize Nokia's SR OS technology.

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

1.2.1 TOE Product 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] and Network Device collaborative

Protection Profile (NDcPP) Extended Package for MACsec Ethernet Encryption, Version 1.2 [MACsec v1.2].

1.3 TOE Description

The TOE portfolio delivers high-performance, scaling and flexibility to support a full array of IP and MPLS services and functions for service provider, web scale and enterprise networks. The 7750 SR Family includes a wide range of physical platforms that share a mutual architecture and feature set. This allows Nokia customers to select the platform that best addresses their unique business goals and fulfills their scale, density, space, power, and value-added service requirements without compromising on quality or features. The 7750 series are chassis-based routers. The TOE supports a full array of network functions and services, achieving scale and efficiency without compromising versatility. It provides highly available service delivery mechanisms that maximize network stability and minimize service interruptions. Every Nokia 7750 series routing appliance is a whole routing system that provides a variety of high-speed interfaces (only Ethernet is within scope of this evaluation) for various scale of networks and various network applications. The TOE utilizes a common Nokia SR OS firmware, features, and technology for compatibility across all platforms.

Nokia SR OS firmware is mainly responsible for all the functionalities and services provided by the routers. The routers can be accessed either via a local console or via a network connection that is protected using the SSH protocol. Each time a user accesses the routers, either via local console terminal connection or from the network remotely using SSH, the user must ensure to successfully authenticate itself with the correct credentials.

The TOE also supports MACsec functionality between compatible Nokia MACsec peer devices using the Media Dependent Adapter (MDA). The TOE permits only EAPOL (PAE EtherType 88-8E), MACsec frames (EtherType 88-E5), and MAC control frames (EtherType is 88-08) and discards others.

The MDAs are pluggable adapter cards. They provide physical interface connectivity to the devices. MDAs can be different in terms of connectivity and density configuration settings. Additionally, the MDA modules vary by chassis. Regardless, they provide the same functionality and security for the related chassis. MDAs support ethernet and multiservice interfaces. For this evaluation, the following is true:

- Routers 7750 SR-a4 and 7750 SR-a8 support 10-port 10/1GE MACsec MDA maxp10-10/1Gb-msec-sfp+
- Routers 7750 SR-1e, 7750 SR-2e, 7750 SR-3e, 7750 SR-7, 7750 SR-12 and 7750 SR-12e support MDA me12-10/1gb-sfp+.

The MACsec Key Agreement (MKA) protocol uses the Connectivity Association Key (CAK) to derive transient session keys called Secure Association Keys (SAKs). SAKs and other MKA parameters are required to sustain communication over the secure channel and to perform encryption and other MACsec security functions. SAKs, along with other essential control information, are distributed in MKA protocol control packets, also referred to as MKPDUs. MACsec can be deployed in two modes:

- Point-to-point mode
- Point-to-multipoint mode

In the evaluated configuration, MACsec is configured individually on a point-to-multipoint Ethernet link. A pair of MACsec devices can be connected via bridge or a direct connection. In order to establish the secured channel, the MACsec devices rely on a Connectivity Association Key (CAK) and utilize the MKA protocol to make and receive the successful secure connection.

In order to determine an authorized peer, both devices must first exchange an MKA frame, and these devices must agree upon a shared key and MACsec cipher suite in order to set up transmit Security Associations (SA). Once the connections are established, the MACsec frames will be transmitted between devices.

The TOE is comprised of the models as indicated in Table 2 below:

Table 2 –TOE Physical Boundary Components

Platform Description	Processors	MACsec MDA
7750 SR-7  # of Cores: 10 Core Frequency: 1.5Ghz OS: Nokia SR OS Image Version: 20.10.R12 Part number: 3HE08423AA	Cavium OCTEON II CN6645	me12-10/1gb-sfp+
7750 SR-12  # of Cores: 10 Core Frequency: 1.5Ghz OS: Nokia SR OS Image Version: 20.10.R12 Part number: 3HE08423AA	Cavium OCTEON II CN6645	me12-10/1gb-sfp+

Platform Description	Processors	MACsec MDA
7750 SR-12e  # of Cores: 10 Core Frequency: 1.5Ghz OS: Nokia SR OS Image Version: 20.10.R12 Part number: 3HE08423AA	Cavium OCTEON II CN6645	me12-10/1gb-sfp+
7750 SR-1e  # of Cores: 10 Core Frequency: 1.3Ghz OS: Nokia SR OS Image Version: 20.10.R12 Part number: 3HE10301AA	Cavium OCTEON II CN6645	me12-10/1gb-sfp+
7750 SR-2e  # of Cores: 10 Core Frequency: 1.3Ghz OS: Nokia SR OS Image Version: 20.10.R12 Part number: 3HE10302AA	Cavium OCTEON II CN6645	me12-10/1gb-sfp+

Platform Description	Processors	MACsec MDA
7750 SR-3e  # of Cores: 10 Core Frequency: 1.3Ghz OS: Nokia SR OS Image Version: 20.10.R12 Part number: 3HE10303AA	Cavium OCTEON II CN6645	me12-10/1gb-sfp+
7750 SR-a4  # of Cores: 6 Core Frequency: 800Mhz OS: Nokia SR OS Image Version: 20.10.R12 Part number: 3HE09195AA	Cavium OCTEON II CN6635	maxp10-10/1Gb-msec-sfp+
7750 SR-a8  # of Cores: 6 Core Frequency: 800Mhz OS: Nokia SR OS Image Version: 20.10.R12 Part number: 3HE09196AA	Cavium OCTEON II CN6635	maxp10-10/1Gb-msec-sfp+

Figure 1 depicts the TOE boundary:

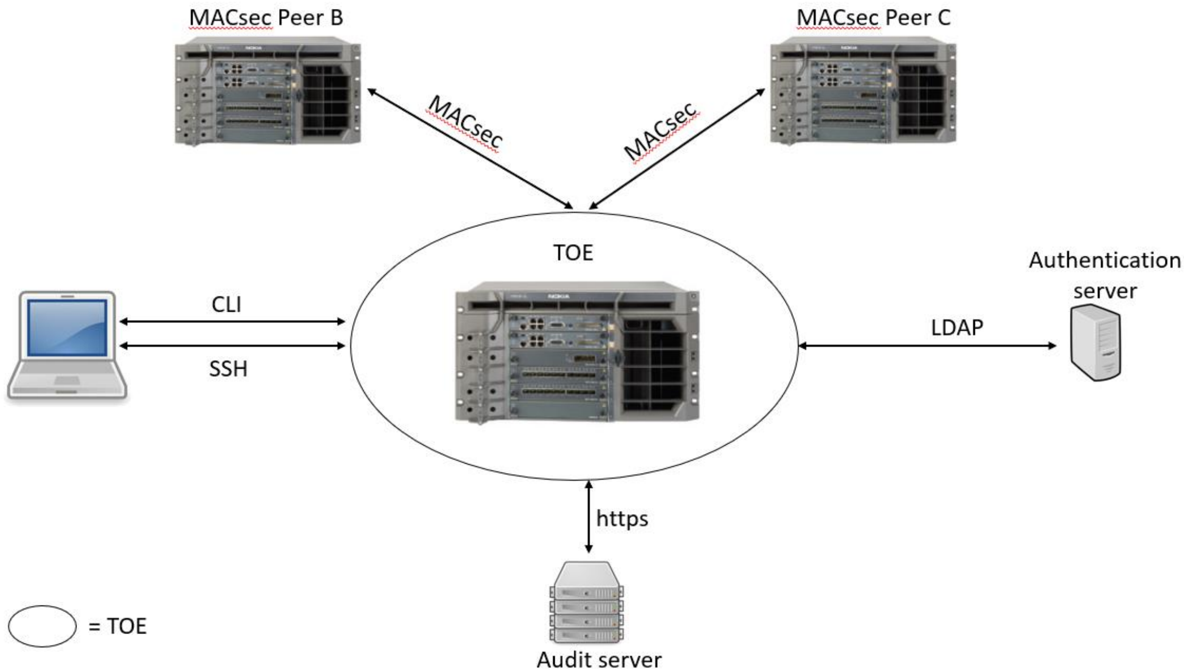

Figure 1 – TOE Boundary Diagram

1.4 TOE Evaluated Configuration

In the evaluated configuration, the TOE consists of the platforms as stated in Section 1.3. The TOE supports secure connectivity with another IT environment device as stated in Table 3.

Table 3 – IT Environment Components

Components	Required (Y/N)	Usage
Audit server	Yes	The audit server supports HTTP PUT requests over TLS v1.2 to receive audit files securely from the TOE.
LDAP server	Yes	This server will provide the authentication mechanism to authenticate users.
MACsec peer	Yes	This peer is required to test the MACsec functionality.
Management workstation with Web Browser/SSH client	Yes	This includes any IT Environment Management workstation with a Web Browser and an SSH client.
Certificate Authority server	Yes	The Certificate Authority server is used for creation and management of X509 certificates to be used with the TOE.

1.5 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 above.

1.6 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.

1.6.1 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 using HTTP PUT requests over TLS v1.2 protocol. The TOE uses a cron script to periodically transfer the audit logs to a URL hosted by the external audit server. Each audit record contains the date and time of event, type of event, subject identity, and the relevant data of the event.

1.6.2 Cryptographic Support

The TOE provides cryptographic support for the services described in Table 4 below. The related CAVP validation details are provided in Table 5. The operating system is SR OS 20.10.R12. The TOE leverages OpenSSL v 1.1.1g library for its cryptographic functionality.

Table 4 – TOE Cryptography Implementation

Cryptographic Method	Usage
FCS_CKM.1 Cryptographic Key Generation	Cryptographic key generation conforming to FIPS PUB 186-4 Digital Signature Standard (DSS), Appendix B.3 and 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

Cryptographic Method	Usage
FCS_CKM.2 Cryptographic Key Establishment	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 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 [groups listed in RFC 3526].
FCS_CKM.4 Cryptographic Key Destruction	Refer to Table 19 for Key Zeroization details.
FCS_COP.1/DataEncryption	AES encryption and decryption conforming to CBC as specified in ISO 10116, CTR as specified in ISO 10116 and GCM as specified in ISO 19772. AES key size supported is 128 bits and 256 bits AES modes supported are: CBC, CTR and GCM.
FCS_COP.1(1)/KeyedHashCMAC Cryptographic Operation (AES-CMAC Keyed Hash Algorithm)	AES CMAC with key sizes 128 bits and 256 bits are supported which meet NIST SP 800-38B.
FCS_COP.1(5) Cryptographic Operation (MACsec AES Data Encryption/Decryption)	AES Key Wrap with key sizes 128 bit and 256 bits are supported which meet NIST SP 800-38F. AES encryption and decryption conforming to GCM as specified in ISO 19772. AES key size supported is 128 bits and 256 bits AES mode supported is: GCM.
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 key 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.
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_HTTPS_EXT.1 HTTPS Protocol	The TOE supports HTTPS protocol that complies with RFC 2818. The TOE implements HTTPS protocol using TLS v1.2 in support of the audit server.
FCS_TLSC_EXT.1 TLS Client Protocol	The TOE supports TLS v1.2 protocol for use with X. 509v3 based authentication. The following ciphersuites in the evaluated configuration: TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268 TLS_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268 TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246 TLS_RSA_WITH_AES_256_CBC_ SHA256 as defined in RFC 5246

Cryptographic Method	Usage
FCS_SSHS_EXT.1 SSH Client Protocol	The TOE supports SSH v2 protocol complaint to the following RFCs: 4251, 4252, 4253, 4254, 4344, 5647, 8268, 6668. The TOE supports public key and password-based authentication. SSH public-key authentication uses ssh-rsa. SSH transport uses the following encryption algorithms: aes128-ctr, aes128-cbc, aes256-cbc and aes256-ctr. Packets greater than 256K bytes in an SSH transport connection are dropped. 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.

Table 5 – CAVP Algorithm Testing References

Cryptographic Algorithms	CAVPS	Implementation Library	Operational Environment (OE)
AES	3969	Microsemi Intellisec 10G PHY (VSC8258)	Microsemi Intellisec 10G PHY
	C2084	Nokia 7x50 SR OS Cryptographic Library	Cavium OCTEON II CN6635
	C2075	Nokia 7x50 SR OS Cryptographic Library	Cavium OCTEON II CN6645
RSA	C2084	Nokia 7x50 SR OS Cryptographic Library	Cavium OCTEON II CN6635
	C2075	Nokia 7x50 SR OS Cryptographic Library	Cavium OCTEON II CN6645
HMAC	C2084	Nokia 7x50 SR OS Cryptographic Library	Cavium OCTEON II CN6635
	C2075	Nokia 7x50 SR OS Cryptographic Library	Cavium OCTEON II CN6645
SHS	C2084	Nokia 7x50 SR OS Cryptographic Library	Cavium OCTEON II CN6635
	C2075	Nokia 7x50 SR OS Cryptographic Library	Cavium OCTEON II CN6645
DRBG	C2084	Nokia 7x50 SR OS Cryptographic Library	Cavium OCTEON II CN6635
	C2075	Nokia 7x50 SR OS Cryptographic Library	Cavium OCTEON II CN6645

1.6.3 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.

1.6.4 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 –Admin and User roles
- Configurable banners to be displayed at login
- Timeouts to terminate administrative sessions after a set period of inactivity
- Protection of secret keys and passwords

1.6.5 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.

1.6.6 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.

1.6.7 Trusted Path/Channels

The TOE supports HTTPS PUT requests over TLS v1.2 for secure communication to the audit server. The TOE supports TLS v1.2 for secure communication to the LDAP server for authentication. The TOE supports local CLI and uses SSH v2 for secure remote administration.

1.7 Excluded Functionality

The following interfaces are not included as part of the evaluated configuration:

- NTP server (optional).
- gRPC is disabled.
- telnet is disabled.
- MPLS is not evaluated.
- SNMP is not evaluated.

1.8 TOE Documentation

The table below lists the TOE guidance documentation. The Common Criteria (CC) guidance document and TOE ST are provided in .pdf form on the NIAP portal.

Table 6 – TOE Documentation

Reference	Title	Version	Date
[CC]	Nokia 7x50 SROS 20.10.R12 Common Criteria Guidance document	0.8	May 30, 2023
[ST]	Nokia 7x50 SR OS 20.10.R12 Security Target	3.4	May 30, 2023

1.9 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]
- Network Device collaborative Protection Profile (NDcPP) Extended Package MACsec Ethernet Encryption, Version 1.2 [MACsec v1.2]

2 Conformance Claims

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

2.1 CC Conformance Claims

This 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: Part 2 extended
- Common Criteria for Information Technology Security Evaluations Part 2, Version 3.1, Revision 5, April 2017: Part 3 extended

2.2 Protection Profile Conformance

This ST claims exact conformance to the following:

- Collaborative Protection Profile for Network Devices, Version 2.2e [NDcPP v2.2e]
- Network Device collaborative Protection Profile (NDcPP) Extended Package MACsec Ethernet Encryption Version 1.2 [MACsec v1.2]

2.3 Conformance Rationale

This TOE claims exact conformance to NDcPP v2.2e and MACsec v1.2. The security problem definition, security objectives and security requirements in this ST are all taken from the relevant Protection Profile and Extended Package, performing only operations defined there.

2.4 Technical Decisions

All NIAP Technical Decisions (TDs) issued to date and applicable to NDcPP v2.2e and MACsec v1.2 have been addressed. Table 7 identifies all TDs relevant to NDcPP v2.2e. Table 8 identifies all TDs relevant to MACsec v1.2.

Table 7 – Technical Decisions for NDcPP v2.2e

Technical Decision	Applicable (Y/N)	Exclusion Rationale (if applicable)
TD0592 – NIT Technical Decision for Local Storage of Audit Records	Yes	
TD0591 – NIT Technical Decision for Virtual TOEs and hypervisors	No	TOE is not virtual.
TD0581 – NIT Technical Decision for Elliptic curve-based key establishment and NIST SP 800-56Arev3	No	TOE does not support ECC certificates.
TD0580 – NIT Technical Decision for clarification about use of DH14 in NDcPPv2.2e	Yes	
TD0572: Restricting FTP_ITC.1 to only IP address identifiers	Yes	
TD0571: Guidance on how to handle FIA_AFL.1.	Yes	
TD0570: clarification about FIA_AFL.1.	Yes	
TD0569: Session ID Usage Conflict in FCS_DTLSS_EXT.1.7	No	DTLSS is not claimed.
TD0564: Vulnerability Analysis Search Criteria.	Yes	
TD0563: Clarification of audit date information	Yes	
TD0556: NIT Technical Decision for RFC 5077 question	No	TLSS is not claimed.
TD0555: NIT Technical Decision for RFC Reference incorrect in TLSS Test	No	TLSS is not claimed.
TD0547: clarification on developer disclosure of software components as part of AVA_VAN.	Yes	
TD0546: DTLS - clarification of Application Note 63	No	DTLS is not claimed.
TD0538: Outdated link to allowed-with list	Yes	
TD0537: Incorrect reference to FCS_TLSC_EXT.2.3	Yes	
TD0536: Update Verification Inconsistency	Yes	
TD0528: Missing EAs for FCS_NTP_EXT.1.4	No	NTP is not claimed .
TD0527: Updates to Certificate Revocation Testing (FIA_X509_EXT.1	No	TOE does not support ECC certificates.

Table 8 – Technical Decisions for MACsec v.1.2

Technical Decision	Applicable (Y/N)	Exclusion Rationale (if applicable)
TD0553: FCS_MACSEC_EXT.1.4 and MAC control frames	Yes	
TD0512: Group CAKs for establishing multiple MKA connections is not mandated (supersedes TD0272)	Yes	
TD0509: Correction to MACsec Audit	Yes	

Technical Decision	Applicable (Y/N)	Exclusion Rationale (if applicable)
TD0487: Correction to Typo in FCS_MACSEC_EXT.4	Yes	
TD0466: Selectable Key Sizes for AES Data Encryption/Decryption	Yes	
TD0273: Rekey after CAK expiration	Yes	
TD0190: FPT_FLS.1(2)/SelfTest Failure with Preservation of Secure State and Modular Network Devices	Yes	
TD0135: SNMP in NDcPP MACsec EP v1.2	No	SNMP is not claimed.
TD0105: MACsec Key Agreement	Yes	

3 Security Problem Definition

The security problem definition has been taken directly from NDcPP v2.2e and MACsec v1.2 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 9 are drawn directly from the [NDcPP v2.2e] and [MACsec v1.2].

Table 9 – Threats

ID	Threat
T.DATA_INTEGRITY	An attacker may modify data transmitted over the MACsec channel in a way that is not detected by the recipient.
T. NETWORK_ACCESS	An attacker may send traffic through the TOE that enables them to access devices in the TOE's Operational Environment without authorization.
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 tunneling 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

ID	Threat
	insert themselves into the network stream and perform a man-in-the-middle attack. The result is the critical network traffic is exposed and there could be a loss of confidentiality and integrity, and potentially the Network Device itself could be compromised.
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 10 are drawn directly from the [NDcPP v2.2e] and [MACsec v1.2]

Table 10 – 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

ID	Assumption
	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.TRUSTED_ADMINISTRATOR	The Security Administrator(s) for the network device are assumed to be trusted and to act in the best interest of security for the organization. This includes being appropriately trained, following policy, and adhering to guidance documentation. Administrators are trusted to ensure passwords/credentials have sufficient strength and entropy and to lack malicious intent when administering the device. The network device is not expected to be capable of defending against a malicious Administrator that actively works to bypass or compromise the security of the device. 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.
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 11 are drawn directly from the [NDcPP v2.2e] and [MACsec v1.2]

Table 11 – 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 from NDcPP v2.2e and MACsec v1.2 and are reproduced here for the convenience of the reader.

4.1 Security Objectives for the TOE

The following security objectives for the operational environment assist the TOE in correctly providing its security functionality. These track with the assumptions about the environment.

Table 12 – Security Objectives

ID	TOE Security Objective
O.CRYPTOGRAPHIC_FUNCTIONS	The TOE will provide cryptographic functions that are used to establish secure communications channels between the TOE and the Operational Environment.
O.AUTHENTICATION	The TOE will provide the ability to establish connectivity associations with other MACsec peers.
O.PORT_FILTERING	The TOE will provide the ability to restrict the flow of traffic between networks based on originating port and established connection information.
O.SYSTEM_MONITORING	The TOE will provide the means to detect when security relevant events occur and generate audit events in response to this detection.
O.AUTHORIZED_ADMINISTRATION	The TOE will provide management functions that can be used to securely manage the TSF.
O.TSF_INTEGRITY	The TOE will provide mechanisms to ensure that it only operates when its integrity is verified.
O.REPLAY_DETECTION	The TOE will provide the means to detect attempted replay of MACsec traffic by inspection of packet header information.
O.VERIFIABLE_UPDATES	The TOE will provide a mechanism to verify the authenticity and integrity of product updates before they are applied.

4.2 Security Objectives for the Operational Environment

The security objectives have been taken from NDcPP v2.2e and are reproduced here for the convenience of the reader. The table below describes the Objectives for the Operational Environment:

Table 13 – 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. Note: For vNDs the TOE includes only the contents of its own VM, and does not include other VMs or the VS.

ID	Objectives for the Operational Environment
OE.TRUSTED_ADMIN	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. Note: For vNDs the TOE includes only the contents of its own VM, and does not include other VMs or the VS.
OE.UPDATES	The TOE firmware and software is updated by an Administrator on a regular basis in response to the release of product updates due to known vulnerabilities.

4.3 Security Objectives Rationale

The Protection Profiles and Extended Packages to which this ST claims conformance are as follows:

- NDcPP v2.2e, Section 5
- MACsec v1.2, Section 2, Section 3, and Appendix A

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, and all international interpretations.

Table 14 – 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_COP.1(1)/KeyedHash:CMAC	Cryptographic Operation (AES-CMAC Keyed Hash Algorithm)
FCS_COP.1(5)	Cryptographic Operation (MACsec AES Data Encryption/Decryption)
FCS_HTTPS_EXT.1	HTTPS Protocol
FCS_MACSEC_EXT.1	MACsec
FCS_MACSEC_EXT.2	MACsec Integrity and Confidentiality
FCS_MACSEC_EXT.3	MACsec Randomness
FCS_MACSEC_EXT.4	MACsec Key Usage
FCS_MKA_EXT.1	MACsec Key Agreement
FCS_RBG_EXT.1	Random Bit Generation
FCS_SSHS_EXT.1	SSH Server Protocol
FCS_TLSC_EXT.1	TLS Client Protocol without Mutual Authentication
FCS_TLSC_EXT.2	TLS Client Protocol with Mutual Authentication
FIA_AFL.1	Authentication Failure Management
FIA_PMG_EXT.1	Password Management
FIA_PSK_EXT.1	Pre-shared Key Composition
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	Certificate Authentication
FIA_X509_EXT.3	Certificate Requests
FMT_MOF.1/Functions	Management of security functions behavior
FMT_MOF.1/ManualUpdate	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_APW_EXT.1	Protection of Administrator Passwords
FPT_CAK_EXT.1	Protection of CAK Data
FPT_FLS.1(2)/SelfTest	Failure with Preservation of Secure State
FPT_RPL.1	Replay Detection

Requirement	Description
FPT_SKP_EXT.1	Protection of TSF Data (for reading of all pre-shared, symmetric and private keys)
FPT_STM_EXT.1	Reliable Time Stamps
FPT_TST_EXT.1	TSF Testing
FPT_TUD_EXT.1	Trusted Update
FTA_SSL_EXT.1	TSF-initiated Session Locking
FTA_SSL.3	TSF-initiated Termination
FTA_SSL.4	User-initiated Termination
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 underlined 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 not specified level of audit; and
c) *All administrative actions comprising.*
   - *Administrative login and logout (name of user account shall be logged if individual user accounts are required for Administrators).*
   - *Changes to TSF data related to configuration changes (in addition to the information that a change occurred it shall be logged what has been changed).*
   - *Generating/import of, changing, or deleting of cryptographic keys (in addition to the action itself a unique key name or key reference shall be logged).*
   - *Resetting passwords (name of related user account shall be logged).*
   - *[no other actions];*
d) *Specifically defined auditable events listed in Table 15*.

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 15*.*

Table 15 – 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_COP.1(1)/KeyedHashCMAC	None	None
FCS_HTTPS_EXT.1	Failure to establish a HTTPS Session.	Reason for failure.
FCS_MACSEC_EXT.1	Session establishment	Secure Channel Identifier (SCI)
FCS_MACSEC_EXT.3.1	Creation and update of Secure Association Key	Creation and update times
FCS_MACSEC_EXT.4.4	Creation of Connectivity Association	Connectivity Association Key Names
FCS_TLSC_EXT.1	Failure to establish a TLS Session	Reason for failure
FCS_TLSC_EXT.2	None	None
FCS_SSHS_EXT.1	Failure to establish an SSH session	Reason for failure
FCS_RBG_EXT.1	None	None
FIA_AFL.1	Unsuccessful login attempts limit is met or exceeded. Administrator lockout due to excessive authentication failures	Origin of the attempt (e.g., IP address)
FMT_MOF.1/Functions	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
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

Requirement	Auditable Events	Additional Audit Record Contents
FMT_MOF.1/ManualUpdate	Any attempt to initiate a manual update	None
FMT_MTD.1/CoreData	None	None
FMT_MTD.1/CryptoKeys	None	None
FMT_SMF.1	None	None
FMT_SMR.2	None	None
FPT_APW_EXT.1	None	None
FPT_RPL.1	Detected replay attempt	None
FPT_SKP_EXT.1	None	None
FPT_STM.1_EXT,1	Discontinuous changes to time - either Administrator actuated or changed via an automated process. (Note that no continuous changes to time need to be logged. See also application note on FPT_STM_EXT.1)	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_TST_EXT.1	None	None
FPT_TUD_EXT.1	Initiation of update; result of the update attempt (success or failure)	None
FTA_SSL_EXT.1 (if "terminate the session is selected)	The termination of a local interactive session by the session locking mechanism.	None
FTA_SSL.3	The termination of a remote session by the session locking mechanism.	None
FTA_SSL.4	The termination of an interactive session.	None
FTA_TAB.1	None	None
FTP_ITC.1	Initiation of the trusted channel. Termination of the trusted channel. Failure of the trusted channel functions.	Identification of the initiator and target of failed trusted channels establishment attempt.
FTP_TRP.1/Admin	Initiation of the trusted path. Termination of the trusted path. 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: *[the oldest log file is overwritten]*] 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 keys in accordance with a specified cryptographic key generation algorithm: [

- *RSA schemes using cryptographic key sizes of 2048-bit or greater that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.3;*
- *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 [groups listed in 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, GCM*] *mode* and cryptographic key sizes [*128 bits, 256 bits*] that meet the following: *AES as specified in ISO 18033-3,* [*CBC as specified in ISO 10116, CTR as specified in ISO 10116, GCM as specified in ISO 19772*].

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_COP.1(1)/KeyedHashCMAC Cryptographic Operation (AES-CMAC Keyed Hash Algorithm)

FCS_COP.1.1(1)/KeyedHash:CMAC Refinement
The TSF shall perform keyed-hash message authentication in accordance with a specified cryptographic algorithm [AES-CMAC] and cryptographic key sizes [128, 256 bits] and message digest size of 128 bits that meets NIST SP 800-38B.

5.2.2.9 FCS_COP.1(5) Cryptographic Operation (MACsec AES Data Encryption/Decryption)

FCS_COP.1.1(5) Refinement
The TSF shall perform **encryption/decryption** in accordance with a specified cryptographic algorithm AES used in **AES Key Wrap, GCM** and cryptographic key sizes **128 bits, 256 bits** that meet the following:

AES as specified in ISO 18033-3, AES Key Wrap as specified in NIST SP 800-38F, GCM as specified in ISO 19772.

5.2.2.10 FCS_HTTPS_EXT.1 HTTPS Protocol

FCS_HTTPS_EXT.1.1
The TSF shall implement the HTTPS protocol that complies with RFC 2818.

FCS_HTTPS_EXT.1.2
The TSF shall implement the HTTPS protocol using TLS.

FCS_HTTPS_EXT.1.3
If a peer certificate is presented, the TSF shall [*not establish the connection*] if the peer certificate is deemed invalid.

5.2.2.11 FCS_MACSEC_EXT.1 MACsec

FCS_MACSEC_EXT.1.1
The TSF shall implement MACsec in accordance with the IEEE Standard 802.1AE-2006.

FCS_MACSEC_EXT.1.2
The TSF shall derive a Secure Channel Identifier (SCI) from a peer's MAC address and port to uniquely identify the originator of a MACsec Protocol Data Unit (MPDU).

FCS_MACSEC_EXT.1.3
The TSF shall reject any MPDUs during a given session that contain an SCI other than the one used to establish that session.

FCS_MACSEC_EXT.1.4
The TSF shall permit only EAPOL (PAE EtherType 88-8E) and MACsec frames (EtherType 88-E5), and MAC control frames (EtherType is 88-08) and shall discard others.

5.2.2.12 FCS_MACSEC_EXT.2 MACsec Integrity and Confidentiality

FCS_MACSEC_EXT.2.1
The TOE shall implement MACsec with support for integrity protection with a confidentiality offset of [0, 30, 50].

FCS_MACSEC_EXT.2.2
The TSF shall provide assurance of the integrity of protocol data units (MPDUs) using an Integrity Check Value (ICV) derived with the Secure Association Key (SAK).

FCS_MACSEC_EXT.2.3
The TSF shall provide the ability to derive an Integrity Check Value Key (ICK) from a CAK using a KDF.

5.2.2.13 FCS_MACSEC_EXT.3 MACsec Randomness

FCS_ MACSEC_EXT.3.1
The TSF shall generate unique Secure Association Keys (SAKs) using [key derivation from Connectivity Association Key (CAK) per section 9.8.1 of IEEE 802.1X-2010] such that the likelihood of a repeating SAK is no less than 1 in 2 to the power of the size of the generated key.

FCS_ MACSEC_EXT.3.2
The TSF shall generate unique nonce for the derivation of SAKs using the TOE's random bit generator as specified by FCS_RBG_EXT.1.

5.2.2.14 FCS_MACSEC_EXT.4 MACsec Key Usage

FCS_MACSEC_EXT.4.1
The TSF shall support peer authentication using pre-shared keys, [no other methods].

FCS_MACSEC_EXT.4.2
The TSF shall distribute SAKs between MACsec peers using AES key wrap as specified in FCS_COP.1(1).

FCS_MACSEC_EXT.4.3
The TSF shall support specifying a lifetime for CAKs.

FCS_MACSEC_EXT.4.4
The TSF shall associate Connectivity Association Key Name (CKN) with ~~CAKs~~ Security Association Key (SAK)s that are defined by the key derivation function using the CAK as input data (per 802.1X, section 9.8.1).

FCS_MACSEC_EXT.4.5
The TSF shall associate Connectivity Association Key Names (CKNs) with CAKs. The length of the CKN shall be an integer number of octets, between 1 and 32 (inclusive).

5.2.2.15 FCS_MKA_EXT.1 MACsec Key Agreement

FCS_MKA_EXT.1.1
The TSF shall implement Key Agreement Protocol (MKA) in accordance with IEEE 802.1X-2010 and 802.1Xbx-2014.

FCS_MKA_EXT.1.2
The TSF shall enable data delay protection for MKA that ensures data frames protected by MACsec are not delayed by more than 2 seconds.

FCS_MKA_ EXT.1.3
The TSF shall provide assurance of the integrity of MKA protocol data units (MKPDUs) using an Integrity Check Value (ICV) derived from an Integrity Check Value Key (ICK).

FCS_MKA_EXT.1.4
The TSF shall provide the ability to derive an Integrity Check Value Key (ICK) from a CAK using a KDF.

FCS_ MKA_EXT.1.5
The TSF shall enforce an MKA Lifetime Timeout limit of 6.0 seconds and MKA Bounded Hello Time limit of 0.5 seconds.

FCS_MKA_EXT.1.6
The Key Server shall refresh a SAK when it expires. The Key Server shall distribute a SAK by [pairwise CAKs]. ~~If group CAK is selected, then the Key Server shall distribute a group CAK by [selection: a group CAK, pairwise CAKs, pre-shared key].~~ If pairwise CAK is selected, then the pairwise CAK shall be [pre-shared key]. The Key Server shall refresh a CAK when it expires.

FCS_MKA_EXT.1.7
The Key Server shall distribute a fresh SAK whenever a member is added to or removed from the live membership of the CA.

FCS_MKA_EXT.1.8
The TSF shall validate MKPDUs according to 802.1X, Section 11.11.2. In particular, the TSF shall discard without further processing any MKPDUs to which any of the following conditions apply:

a) The destination address of the MKPDU was an individual address.

b) The MKPDU is less than 32 octets long.
c) The MKPDU is not a multiple of 4 octets long.
d) The MKPDU comprises fewer octets than indicated by the Basic Parameter Set body length, as encoded in bits 4 through 1 of octet 3 and bits 8 through 1 of octet 4, plus 16 octets of ICV.
e) The CAK Name is not recognized.

If an MKPDU passes these tests, then the TSF will begin processing it as follows:

a) If the Algorithm Agility parameter identifies an algorithm that has been implemented by the receiver, the ICV shall be verified as specified in IEEE 802.1x Section 9.4.1.
b) If the Algorithm Agility parameter is unrecognized or not implemented by the receiver, its value can be recorded for diagnosis but the received MKPDU shall be discarded without further processing.

Each received MKPDU that is validated as specified in this clause and verified as specified in 802.1X, section 9.4.1 shall be decoded as specified in 802.1X, section 11.11.4.

5.2.2.16 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.

5.2.2.17 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, [4344, 8268, 6668].

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

FCS_SSHS_EXT.1.3
The TSF shall ensure that, as described in RFC 4253, packets greater than *[256K]* 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.

5.2.2.18 FCS_TLSC_EXT.1 TLS Client Protocol without Mutual Authentication

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

[

- *TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268*
- *TLS_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268*
- *TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246*
- *TLS_RSA_WITH_AES_256_CBC_ SHA256 as defined in RFC5246*

] and no other ciphersuites.

FCS_TLSC_EXT.1.2
The TSF shall verify that the presented identifier matches [*the reference identifier per RFC 6125 section 6, IPv4 address in SAN, IPv6 address in the SAN*].

FCS_TLSC_EXT.1.3
When establishing a trusted channel, by default the TSF shall not establish a trusted channel if the server certificate is invalid. The TSF shall also [

- *Not implement any administrator override mechanism*

].

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

5.2.2.19 FCS_TLSC_EXT.2 TLS Client Support for Mutual Authentication

FCS_TLSC_EXT.2.1
The TSF shall support TLS communication with mutual authentication using X.509v3 certificates.

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*].

Application Note: This SFR has been copied directly from NDcPP because it is more specific and recent compared to MACsec EP.

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: ["!", "@", "#", "$", "%", "^", "&", "*", "(", ")", *[no other characters]*];

b) Minimum password length shall be configurable to between [*6*] and [*50*] characters.

5.2.3.3 FIA_PSK_EXT.1 Extended: Pre-Shared Key Composition

FIA_PSK_EXT.1.1
The TSF shall use pre-shared keys for MKA as defined by IEEE 802.1X, [no other protocols].

FIA_PSK_EXT.1.2
The TSF shall be able to [accept] bit-based pre-shared keys.

5.2.3.4 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.

5.2.3.5 FIA_UAU_EXT.1 Password-based Authentication Mechanism

FIA_UAU_EXT.2.1
The TSF shall provide a local [*password-based, [LDAP]*] authentication mechanism to perform local administrative user authentication.

5.2.3.6 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**.

5.2.3.7 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.

5.2.3.8 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 [*TLS*] 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*].

5.2.3.9 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*].

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 Behavior

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_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.*

5.2.4.4 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*.

5.2.4.5 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;*
- *Generate a PSK-based CAK and install it in the device.*
- *Manage the Key Server to create, delete, and activate MKA participants [as specified in 802.1X, sections 9.13 and 9.16 (cf. MIB object ieee8021XKayMkaParticipantEntry) and section.12.2 (cf. function createMKA())]*
- *Specify a lifetime of a CAK*
- *Enable, disable, or delete a PSK-based CAK using [[CLI management command]]*
- *Configure the number of failed administrator authentication attempts that will cause an account to be locked out*
- [
    - *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 configure the cryptographic functionality;*
    - *Ability to configure thresholds for SSH rekeying;*
    - *Ability to set the time which is used for time-stamps;*
    - *Ability to manage the cryptographic keys;*
    - *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;*
    - *Configure the time interval for administrator lockout due to excessive authentication failures*
    - *].*

5.2.4.6 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_CAK_EXT.1 Protection of CAK Data

FPT_CAK_EXT.1.1
The TSF shall prevent reading of CAK values by administrators.

5.2.5.3 FPT_FLS.1(2)/SelfTest Failure with Preservation of Secure State

FPT_FLS.1(2)/SelfTest Refinement
The TSF shall **shut down** when any of the following types of failures occur: **failure of the power-on self-tests, failure of integrity check of the TSF executable image, failure of noise source health tests.**

5.2.5.4 FPT_RPL.1 Replay Detection

FPT_RPL.1.1
The TSF shall detect replay for the following entities: [*MPDUs, MKA frames*].

FPT_RPL.1.2
The TSF shall perform [*discarding of the replayed data, logging of the detected replay attempt*] when replay is detected.

5.2.5.5 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.

5.2.5.6 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*].

5.2.5.7 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), periodically at the conditions [BIOS checks, cryptographic library functionality test, and firmware integrity checks]*] to demonstrate the correct operation of the TSF: [

- *Integrity Test*
- *AES Known Answer Test*
- *CMAC Known Answer Test*
- *GCM Known Answer Test*
- *CCM Known Answer Test*
- *HMAC-SHA-1/256/384/512 Known Answer Test*
- *SHA-1/256/512 Known Answer Test*
- *RSA Signature Known Answer Test*
- *DRBG Known Answer Test*
- *Noise Source Health Test*

].

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

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

FTP_ITC.1.3
The TSF shall initiate communication via the trusted channel for *[audit server communications, LDAP server, and MACsec peers]*.

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 trusted communication channel 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 and any relevant EPs/Modules/Packages contain(s) 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 and any relevant EPs/Modules/Packages, which is/are derived from Common Criteria Version 3.1, Revision 5. The assurance requirements are summarized in the Table 16.

Table 16 – Security Assurance Requirements

Assurance Class	Assurance Components	Component Description
Security Target	ASE_CLL.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
Development	ADV_FSP.1	Basic functionality specification
Guidance Documents	AGD_OPE.1	Operational user guidance
	AGD_PRE.1	Preparative user guidance
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 to satisfy the assurance requirements. The following table lists the details.

Table 17 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

SAR Component	How the SAR will be met
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 Specifications

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

Table 18 – TOE Summary Specification SFR Description

TOE SFR	Rationale
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 15.
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 using HTTPS over TLS v1.2. A cron script can be executed on the TOE to periodically transfer the log files from local storage to the external audit server. The cron job script can be configured from 15 minutes to 1 hour or on a weekly basis. The cron script includes a URL of the external audit sever. The TOE performs the transmission of logs periodically using a cron script. The TOE can rollover from one log file to the next log file based on rollover time. For the TOE to successfully create a log file, the compact flash disk must have a minimum of 10% or 5MB of free space. The TOE is designed to store 6.8 GB records in compact flash drive. When the local storage space for audit data is full, the TOE will overwrite the oldest log file.

<table>
<tr><th>TOE SFR</th><th>Rationale</th></tr>
<tr><td></td><td>The TOE allows to create/manage administrators with different privileges. Some available options to configure such administrators are: “user profile membership”, “grant/deny a user access permission for console ftp grpc li netconf or snmp”, “restrict user to home directory”.

Only Authorized Administrators can access the audit events and have the ability to clear the audit events. The TOE does not allow non-privileged administrators to modify the audit records that are stored locally on the device.</td></tr>
<tr><td>FCS_CKM.1</td><td>To support the cryptographic protocols, the TOE uses RSA schemes using cryptographic key sizes of 2048-bit that meet the following: FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Appendix B.3. The TOE supports 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. The TOE supports DHG14 and DHG16 key generation in support of DH key exchanges as part of SSH.

For both TLS and SSH communications, the RSA keys are used in support of digital signatures.

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 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”. The TOE supports 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 groups listed in RFC 3526.

<table>
<tr><td>Scheme</td><td>SFR</td><td>Services</td></tr>
<tr><td>FFC/DHG14/DHG16</td><td>FCS_SSHS_EXT.1</td><td>Administration</td></tr>
<tr><td rowspan="3">RSA</td><td>FCS_SSHS_EXT.1</td><td>Administration</td></tr>
<tr><td>FCS_HTTPS_EXT.1</td><td>Audit server</td></tr>
<tr><td>FCS_TLSC_EXT.1</td><td>LDAP server</td></tr>
</table>

The relevant NIST CAVP certificate numbers are listed in 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.
Non-volatile SSH keys can be zeroized by deleting the key using the file delete command:
/file delete <path>
SSH keys are only stored persistently in cf3:/ssh if “preserve key” is enabled. It is disabled by default.
The CAK is deleted from RAM whenever the key entry is removed from CLI (“no cak”) command. The CAK is deleted from the Compact Flash when followed by the “/admin save” command.
</td></tr>
</table>

<table>
<tr><th>TOE SFR</th><th>Rationale</th></tr>
<tr><td></td><td>Please refer to Table 19 TOE Zeroization.</td></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 AES key sizes supported are 128 bits and 256 bits and the AES modes supported are: CBC, CTR and GCM. The TOE provides AES encryption and decryption in support of SSHv2 for secure communications.

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.

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 and TLS connections for secure communications.

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 5.</td></tr>
<tr><td>FCS_COP.1(1)/KeyedHashCMAC Cryptographic Operation (AES-CMAC Keyed Hash Algorithm)</td><td>The TOE supports keyed-hash message authentication in accordance with AES-CMAC algorithm conforming to NIST SP 800-38B, The key sizes supported are 128 bits and 256 bits and the message digest size supported is 128 bits.</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 TLS and SSH sessions.

The relevant NIST CAVP certificate numbers are listed in Table 5.</td></tr>
</table>

TOE SFR	Rationale
FCS_COP.1.1(5) Refinement	The TOE performs encryption/decryption in accordance with AES algorithm used in AES Key Wrap, GCM and cryptographic key sizes 128 bits, 256 bits that meet the following: AES as specified in ISO 18033-3, AES Key Wrap as specified in NIST SP 800-38F, GCM as specified in ISO 19772.
FCS_HTTPS_EXT.1	The TOE operates as a (HTTP) client as specified in Section 2, of RFC 2818, to provide a secure means of file transfer. The TOE implements HTTPS using TLS. If a peer certificate is presented, the TOE will not establish the connection if the peer certificate is deemed invalid.
FCS_SSHS_EXT.1	The TOE implements SSH protocol that complies with RFC(s) 4251, 4252, 4253, 4254, 4344, 8268 and 6668. The TOE supports public key authentication and password-based authentication. The following public key algorithms are supported: ssh-rsa. TOE also supports LDAP as an authentication server while using the password-based authentication. This list conforms to FCS_SSHS_EXT.1.5. The SSH client's public key is compared to an authorized keys file which is stored on the TOE. The TOE ensures that packets greater than 256K bytes in an SSH transport connection are dropped as described in RFC 4253. When the TOE detects packets greater than 256K, the connection is disconnected. The TOE supports the following encryption algorithms: aes128-cbc, aes256-cbc, aes128-ctr, and aes256-ctr for SSH transport. There are no optional characteristics specified for FCS_SSHS_EXT.1.4. This list is identical to those claimed for FCS_SSHS_EXT.1.4. The following public key algorithms are supported: ssh-rsa. There are no optional characteristics specified for FCS_SSHS_EXT.1.5. This list is identical to those claimed for FCS_SSHS_EXT.1.5. The TOE supports the following data integrity MAC algorithms: hmac-sha1, hmac-sha2-256, hmac-sha2-512. This list corresponds to the list in FCS_SSHS_EXT.1.6. The TOE supports diffie-hellman-group-14-sha1, diffie-hellman-group-14-sha256 and diffie-hellman-group-16-sha512. This list corresponds to the list in FCS_SSHS_EXT.1.7. The TOE is capable of rekeying. The TOE verifies the following thresholds: ● No longer than one hour ● 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.
FCS_TLSC_EXT.1 FCS_TLSC_EXT.2	The TOE implements TLS v1.2 (RFC 5246) and rejects all other TLS and SSL versions. The TOE supports TLS communication with mutual authentication using X.509v3 certificates. The TOE supports the following ciphersuites: • TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268 • TLS_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268 • TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246 • TLS_RSA_WITH_AES_256_CBC_ SHA256 as defined in RFC 5246 The cipher suites specified are those listed in FCS_TLSC_EXT.1.

<table>
<tr><th>TOE SFR</th><th>Rationale</th></tr>
<tr><td></td><td>The TOE does not present the Supported Elliptic Curves/Supported Groups Extension in the Client Hello.

TLS is used for HTTPS/TLS for management purposes and to establish encrypted sessions with other instances of the TOE and IT entities to send/receive audit data.
The TOE verifies that the presented identifier matches the reference identifiers. The TOE supports reference identifiers according to RFC 6125, Section 6, which includes DNS-ID and CN-ID, IPv4 address in SAN, and IPv6 address in the SAN. The TOE supports wild cards. The TOE does not support certificate pinning.
When presented with X509 certificates, 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:
    ○ 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.
    ○ 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.

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 establishing a trusted channel, by default the TOE will not establish a trusted channel if the server certificate is invalid. The TOE does not implement any administrator override mechanism.

The use of CRL is configurable and can be used for certificate revocation. If the TOE is unable to establish a connection to determine the validity of a certificate, the TOE will not accept the certificate.</td></tr>
<tr><td>FCS_MACSEC_EXT.1</td><td>The TOE implements MACsec in accordance with IEEE Standard 802.1AE-2006.
The TOE conforms to IEEE 802.1AE-2006, Sections 5.3a-h, 5.3j-q, 5.4e-h, 8, 9, 10.5,10.6 and 14; IEEE 802.1AEbw-2013 sections 8, 9, 10 and 14.
The MACsec connections preserve confidentiality of communicated data and act to protect against frames that are wrongly transmitted.

The TOE receives SCI based on the port MAC address and sub-port/VLAN ID (1 to 1023). The TOE rejects data with an incorrect SCI value. The TOE permits only EAPOL (PAE EtherType 88-8E), MACsec frames (EtherType 88-E5), and MAC control frames (EtherType is 88-08) and discards others.</td></tr>
<tr><td>FCS_MACSEC_EXT.2</td><td>The TOE provides integrity protection by limiting confidentially offsets to 0, 30 and 50 values.</td></tr>
</table>

<table>
<tr><th>TOE SFR</th><th>Rationale</th></tr>
<tr><td></td><td>The TOE derives the ICV from a CAK using KDF, using the SCI as the most significant bits of the IV and the 32 least significant bits of the PN as the IV. The supported ICV length is 16 octets.

An ICV derived with the SAK is used to provide assurance of the integrity of MPDUs.
The ICV is generated in 2 modes:
• With the compliance of 802.1AE, L2 MAC is in clear and all other bytes are encrypted and also part of the ICV calculation.
• When VLAN is clear, 802.1q tags are not the part of the ICV calculation.</td></tr>
<tr><td>FCS_MACSEC_EXT.3</td><td>The TOE supports CAK of 32 hex characters for aes-128-cmac encryption algorithm and 64 hex characters for aes-256-cmac encryption algorithm.

The TOE supports CAK, which is based on AES cipher in CMAC mode and key sizes of 128 and 256 bits. Each of the keys used by MKA is derived from the CAK. When the TOE uses AES 128-bit CMAC mode encryption, the supported key string is 32-bit hexadecimal in length. When the TOE uses 256-bit encryption, the supported key size is 64-bit hexadecimal in length.

SAKs are generated using the Key Server’s RNG function. The TOE generates unique Secure Association Keys (SAKs) using key derivation from Connectivity Association Key (CAK) per section 9.8.1 of IEEE 802.1X-2010 such that the likelihood of a repeating SAK is no less than 1 in 2 to the power of the size of the generated key.
The TOE generates a unique nonce for the derivation of SAKs using the TOE’s random bit generator as specified by FCS_RBG_EXT.1.</td></tr>
<tr><td>FCS_MACSEC_EXT.4</td><td>The TOE ensures MACsec peer authentication by only using pre-shared keys.

The TOE uses AES Key Wrap to distribute the SAKs between peers.
The TOE supports aes-128-cmac or aes-256-cmac for SAK Wrapping.</td></tr>
<tr><td>FCS_MKA_EXT.1</td><td>The TOE supports Key Agreement Protocol (MKA) in agreement with standard IEEE 802.1X-2010 and 802.1Xbx-2014.

The TOE implements MKA Lifetime Timeout limit of 6 to 18 seconds and a Hello Timeout limit of 2 sec with Hello Timeout configuration values of 500ms, and 1 to 6 sec.

The TOE supports the data delay protection to provide security against delay attacks.

The TOE allows for the configuration of the replayWindow size. For any encrypted data packets arriving with a PN in the SecTag, if the PN falls out of the replayWindow, it gets dropped and LatePkt counter is incremented.

The TOE runs assurance of the integrity of MKA protocol data units using an ICV derived from the ICK. The ICK is derived from the CAK as per IEEE 802.1X-2010, Section 9.3.3 derived keys using AES-CMAC.

The ICV is checked on the reception of each MKA PDU.

The TOE refreshes the SAK in the following situations:
• When a live peer leaves the CA or a new host has joined the CA domain and becomes a member.</td></tr>
</table>

TOE SFR	Rationale
	• When PN (Packet Number) reaches 0xc0000000 (or XPN 0xc000000000000000) a new SAK is generated in order to avoid PN exhaustion. • When a new PSK is configured and a rollover of PSK has been executed. The TOE supports pairwise CAK. Additionally, the TOE discards the MKPDU with individual addresses. The TOE checks the MKPDU length and discards MKPDUs smaller than 32 octets. The TOE also verifies and discards MKPDUs that are not multiples of 4 octets long. The TOE verifies the size as required by the basic parameter set body length and discards MKPDUs with unrecognized CKN. If an MKPDU passes these tests, then the TOE will begin processing it as follows: a) If the Algorithm Agility parameter identifies an algorithm that has been implemented by the receiver, the ICV shall be verified as specified in IEEE 802.1X Section 9.4.1. b) If the Algorithm Agility parameter is unrecognized or not implemented by the receiver, its value can be recorded for diagnosis but the received MKPDU shall be discarded without further processing.
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 a 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-based noise sources. The CTR_DRBG is seeded with a minimum of 256 bits of entropy.
FPT_RPL.1	The TOE discards the replayed data, and replay data is logged by the TOE. When the TOE receives a valid MKA PDU (size and ICV size checks pass), the Message Number (MN) is checked to ensure that it is greater than the previous MN received from this peer. If the MN is not greater than the previous MN, logs are generated. If the situation persists, then the MKA operational state will switched to off based on the MKA timeout. Additionally, the attempt to replay data is logged.
FPT_FLS.1(2)/ SelfTest	When a failure occurs within the TOE (e.g., power-on self-tests or integrity check of the TOE executable image tests), it securely disables its interfaces and then reboots. When the TOE boots in FIPS mode, it runs a diagnostic against the FIPS module and algorithms. Any failure in this self-diagnostic will cause the TOE to reboot.
FIA_AFL.1/NDcPP and MACsec	The Security Administrator can configure the maximum number of failed attempts for the CLI interface. The TOE allows the administrator to configure the number of successive failed authentication attempts. When a user fails to authenticate a number of times equal to the configured limit, the TOE locks the claimed user identity until the configured time is reached. Once the elapsed time has passed, the user will be able re-login. Administrators can configure unsuccessful authentication attempts range between 1 – 64 within a configurable time limit of 0 to 60 minutes. When the account is locked, the TOE does not permit any further actions until the account is accessible.

TOE SFR	Rationale
	The authentication failures cannot lead to a situation where no administrator access is available. A user would be configured to access the LDAP server which would provide local access to the TOE. The LDAP server is not subject to lockout.
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 is configurable to between 6 to 50 characters.
FIA_UIA_EXT.1 FIA_UAU_EXT.2	The TOE does not permit any actions prior to Administrators logging into the TOE. They are able to view the banner at the login prompt. 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. The TOE supports RSA public key authentication as a server, password-based authentication for remote and local authentication, and LDAP authentication for remote and local users. 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.
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 TLS 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, Certificate Revocation List (CRL) as specified in RFC 5759 Section 5
	• The TOE validates the extendedKeyUsage field according to the following rules:
	  ○ 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.
	  ○ 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.

TOE SFR	Rationale
	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. 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.
FIA_X509_EXT.2	X.509 certificate can be used to authenticate and establish secure communication channel for LDAP server. RSA based certificates: The supported RSA key size shall be 2048 bits. When establishing a connection and the TOE cannot determine the validity of a certificate, the TOE will not accept the certificate. To validate a peer certificate on the TOE, an authenticated administrator must import its CA certificates and CRLs. CA profiles must be created and enabled for each imported CA certificate and CRL. The administrator must configure at least one trust anchor to limit the list of CA certificates. Furthermore, the administrator can create a client profile to specify the cipher-list and client certificate to use.
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 and Country. 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.
FPT_CAK_EXT.1	The TOE stores CAKs in an encrypted form. This prevents the CAK value from being displayed in clear text to the administrators on the CLI. The TOE uses AES-256 to encrypt and protect the CAKs.
FIA_PSK_EXT.1.	The TOE supports the use of pre-shared keys for MACsec key agreement protocols as defined by IEEE 802.1X. The pre-shared keys are not generated by the TOE but rather the TOE will accept bit based pre-shared keys. The CAK in PSK can be configured as follow: 128-bit CAK is 32 Hex digits, 256-bit CAK is 64 hex digits.
FMT_MOF.1/Functions	The TOE restricts the ability to modify the behaviour of transmission of audit data to an audit server to Security Administrators.
FMT_MOF.1/ManualUpdate	Security Administrators have the ability to query the current version of the TOE and they are able to perform manual software updates. The currently active version of the TOE can be queried by issuing the “show version” command. The TOE provides means to authenticate firmware updates to the TOE using a published hash prior to installing the firmware. Customers must log in to https://customer.nokia.com/support/s/ portal to download software updates, and then copy it onto the compact flash. The compact flash is then

<table>
<tr><th>TOE SFR</th><th>Rationale</th></tr>
<tr><td></td><td>inserted into the standby CPM. Customers receive a compact flash with the updated software or are instructed to copy a downloaded image onto a compact flash (received out of band).
The compact flash is inserted into the standby CPM and then plugged into the chassis.

When the standby CPM boots, its bootloader will extract the published hmac-sha256 hash from a file in the compact flash and compare it with the hmac-sha256 hash computed over the new software binary. If the hashes match, then it will "jump" into or run the new software binary. Otherwise, it will show a FIPS HMAC-SHA256 error on the console, reboot and repeat the cycle. Meanwhile, the active CPM in the same chassis is still running the current software. Once it detects the standby CPM is operational with the new updated software, then the operator has the option to switchover to the standby CPM running the authenticated software. This switchover mechanism provides minimal service interruption during a software upgrade for our customers. The administrator must authorize the switchover to the standby CPM.</td></tr>
<tr><td>FMT_MTD.1/CoreData</td><td>The TOE implements Role Based Access Control (RBAC). Security Administrative must login before they can access any administrative functions. Only administrators can manage the certificates in TOE's trust store.

The TOE maintains the following roles: Admin and User. Each role defined has a set of permissions that will grant them access to the TOE data. The only interfaces available to an unauthenticated user are the TOE login prompts. Only authorized security administrators may authenticate to the TOE and interact with TSF data. The TOE prevents non-security administrators from modifying any TSF element or security function.

There are two types of trust stores in the TOE: Active and Inactive. In volatile memory, the trust store is active, and the other inactive trust store resides on the persistent store inform of files. The Administrator can assign privileges to non-administrative user by configuring the capabilities in specifically user's profile.</td></tr>
<tr><td>FMT_MTD.1/CryptoKeys</td><td>The Security Administrator has the ability to configure the pre-shared key for MACsec functionality and can modify, generate, and delete the key for SSH.

The TOE restricts the ability to manage SSH (session keys), TLS (session keys), and any configured X.509 certificates (public and private key pairs) to security administrators via command line.</td></tr>
<tr><td>FMT_SMF.1</td><td>The TOE supports the following roles: Administrator. The TOE can be accessed via local CLI and remote 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] capability prior to installing those updates;
Ability to configure the authentication failure parameters for FIA_AFL.1;
Generate a PSK-based CAK and install it in the device.
Manage the Key Server to create, delete, and activate MKA participants [as specified in 802.1X, sections 9.13 and 9.16 (cf. MIB object</td></tr>
</table>

TOE SFR	Rationale
	ieee8021XKayMkaParticipantEntry) and section.12.2 (cf. function createMKA())] • Specify a lifetime of a CAK • Enable, disable, or delete a PSK-based CAK using [[CLI management command]] • Configure the number of failed administrator authentication attempts that will cause an account to be locked out • 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 configure the cryptographic functionality • Ability to configure thresholds for SSH rekeying • Ability to set the time which is used for time-stamps • Ability to manage the cryptographic keys • 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 • Configure the time interval for administrator lockout due to excessive authentication failures
FMT_SMR.2	Security Administrators can configure user's privilege that grant or deny access to TSF data and functions. The Security Administrator can also configure the user's profile to set the following restrictions: **Functionality** — **Description** cli-session-gr — To Add/remove cli-session-group the profile belongs to combined-max-s — To define Maximum number of concurrent SSH & Telnet sessions default-action — To get Default action for the profile entry — To find the Match criteria entry for the profile grpc — To view the gRPC specific profile netconf — To config the netconf specific profile ssh-max-session — To create Maximum number of concurrent SSH sessions The TOE enables both local console access and remote access via SSHv2 secure connection.
FPT_SKP_EXT.1	The TOE stores all private keys in a secure storage that is not accessible through an interface to administrators.
FPT_APW_EXT.1	All passwords are stored in a secure directory that is not readily accessible to administrators. The TOE stores passwords as non-reversible hashes.
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 periodically until the CF is replaced with an authentic firmware.

TOE SFR	Rationale
	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: • 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 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 plaintext is encrypted using AES-GCM with a known 256-bit key, and the computed ciphertext is compared to the expected ciphertext. If they match, then the computed ciphertext is decrypted using the same key, and the recovered plaintext is compared with the original known plaintext. 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 ciphertext is decrypted using the same key, and the recovered plaintext is compared with the original known plaintext. 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.

TOE SFR	Rationale
	• 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	Security Administrators have the ability to query the current version of the TOE and they are able to perform manual software updates. The currently active version of the TOE can be queried by issuing the "show version" command. The TOE provides means to authenticate firmware updates to the TOE using a published hash prior to installing the firmware. Customers must log in to https://customer.nokia.com/support/s/ portal to download software updates, and then copy it onto the compact flash. The compact flash is then inserted into the standby CPM. Customers receive a compact flash with the updated software or are instructed to copy a downloaded image onto a compact flash (received out of band). The compact flash is inserted into the standby CPM and then plugged into the chassis. When the standby CPM boots, its bootloader will extract the published hmac-sha256 hash from a file in the compact flash and compare it with the hmac-sha256 hash computed over the new software binary. If the hashes match, then it will "jump" into or run the new software binary. Otherwise, it will show a FIPS HMAC-SHA256 error on the console, reboot and repeat the cycle. Meanwhile, the active CPM in the same chassis is still running the current software. Once it detects the standby CPM is operational with the new updated software, then the operator has the option to switchover to the standby CPM running the authenticated software. This switchover mechanism provides minimal service interruption during a software upgrade for customers. The administrator must authorize the switchover to the standby CPM.
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.
FTA_SSL_EXT.1 FTA_SSL.3	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 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.

TOE SFR	Rationale
	The allowable inactivity timeout range is from 1 to 1440 minutes.
FTA_SSL.4	The Security Administrator is able to terminate their CLI.
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 LDAP server. The TOE protects communications with an external audit server using HTTPS over TLS v1.2 protocol. The TOE protects communications with an LDAP server using TLS v1.2 protocol. The TOE uses TLS v1.2 protocol with X.509 certificate-based authentication. The TOE secures the communication between its peers using the MACsec at Layer 2. 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. Each SSH v2.0 session is encrypted using AES to protect confidentiality and uses HMACs to protect the integrity of traffic. The protocols listed are consistent with those specified in the requirement.

7 Cryptographic Key Destruction

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

Table 19 – Key Zeroization

Keys/CSPs	Purpose	Storage Location	Method of Zeroization
Diffie-Hellman Shared Secret	The shared secret used in Diffie-Hellman (DH) exchange. Created per the Diffie-Hellman Exchange.	RAM	A single overwrite consisting of zeroes.
Diffie Hellman private key	The private key used in Diffie-Hellman (DH) Exchange	RAM	A single overwrite consisting of zeroes.
SSH Private key	The SSH server host private key is stored on the local filesystem	RAM; Compact Flash if preserve-key is enabled.	A single overwrite consisting of zeroes. Non-volatile SSH keys can be zeroized by deleting the key using the file delete command: /file delete <path>
SSH Session Key	These are the session keys for SSH.	RAM	A single overwrite consisting of zeroes.
TLS Session Keys	These are the session keys for TLS.	RAM	A single overwrite consisting of zeroes.
MACsec Security Association Key (SAK)	The SAK is used to secure the control plane traffic.	RAM	A single overwrite consisting of zeroes.
MACsec Connectivity Association Key (CAK)	The CAK secures the traffic.	RAM and Compact Flash.	A single overwrite consisting of zeroes. The CAK is deleted from RAM whenever the key entry is removed from CLI (“no cak”) command. The CAK is deleted from the Compact Flash when followed by the “/admin save” command.
MACsec Key Encryption Key (KEK)	The Key Encrypting Key (KEK) is used by Key Server, elected by MKA, to transport a succession of SAKs, for use by MACsec, to the other member(s) of a Secure Connectivity Association (CA).	RAM	A single overwrite consisting of zeroes.
MACsec Integrity Check Key (ICK)	The ICK is used to verify the integrity of MPDUs and	RAM	A single overwrite consisting of zeroes.

Keys/CSPs	Purpose	Storage Location	Method of Zeroization
	to prove that the transmitter of the MKPDU possesses the CAK.		
RNG Seed Key	This is the seed key for the RNG.	RAM	A single overwrite consisting of zeroes.
RNG Seed	This seed is for the RNG.	RAM	A single overwrite consisting of zeroes

8 Acronym Table

Acronyms should be included as an Appendix in each document.

Table 20 – Acronyms

Acronym	Definition
AES	Advanced Encryption Standard
ARP	Address Resolution Protocol
ASCII	American Standard Code for Information Interchange
BIOS	Basic Input/Output System
CAK	Connectivity Association Key
CCRA	Arrangement on the Recognition of Common Criteria Certificates in the field of IT Security
CBC	Cipher Block Chaining
CLI	Command Line Interface
CRL	Certificate Revocation List
CSR	Certificate Signing Request
DH	Diffie-Hellman
DHE	Diffie-Hellman Ephemeral
DNS	Domain Name System
DRBG	Deterministic Random Bit Generator
DSA	Digital Signature Algorithm
FIPS	Federal Information Processing Standards
GCM	Galois Counter Mode
gRPC	gRPC Remote Procedure Calls
GUI	Graphical User Interface
HMAC	Hash Message Authentication Code
HTTP	Hypertext Transfer Protocol
HTTPs	Hypertext Transfer Protocol Secure
IP	Internet Protocol
KAT	Known Answer Test
KDF	Key Derivation Function
LDAP	Lightweight Directory Access Protocol
MB	Megabyte
MKA	MACsec Key Agreement
NIAP	National Information Assurance Partnership
NTP	Network Time Protocol
OSP	Organizational Security Policy
PCT	Pairwise Consistency Test
PP	Protection Profile
PKCS	Public Key Cryptography Standards
RAM	Random Access Memory
RFC	Requests for Comments

Acronym	Definition
RSA	Rivest-Shamir-Adleman
SAR	Security Assurance Requirement
SFR	Security Functional Requirement
SFP	Security Policy Database
SHA	Secure Hash Algorithm
SNMP	Simple Network Management Protocol
SSH	Secure Shell
SSO	Single Sign On
ST	Security Target
TOE	Target of Evaluation
TLS	Transport Layer Security
TSF	TOE Security Functionality
TSFI	TSF Interface
TSS	TOE Summary Specification
UI	User Interface
URI	Uniform Resource Identifier
MPLS	Multiprotocol Label Switching