ciena
Experience. Outcomes.

Title:	Security Target for the Ciena 6500 Packet-Optical Platform Flex3 WaveLogic 3e OCLD Encryption Module
Doc-ID	323-1851-CC-ST-NTK539QS-NTK539QV
ST Version:	1.6
Date:	April 28, 2020
Author:	Ciena Corporation 7035 Ridge Road Hanover, MD 21076
Certification-ID:	BSI-DSZ-CC-1082

Table of Contents

Figures

Tables

1 Security Target Introduction

The scope of this Security Target is to describe the Ciena 6500 Packet-Optical Platform Flex3 WaveLogic 3e OCLD Encryption Module in terms of Common Criteria and to specify its security functional and security assurance requirements.

1.1 ST Reference

The ST reference provides identification attributes for the ST.

ST Title:	Security Target for the Ciena 6500 Packet-Optical Platform Flex3 WaveLogic 3e OCLD Encryption Module
ST Version:	1.6
ST Date:	April 28, 2020
ST Authors:	Ciena Corporation
CC Version:	Version 3.1 Revision 5
Assurance Level:	EAL2 augmented by ALC_FLR.2 (hereafter called „EAL2+“)
Certification ID:	BSI-DSZ-CC-1082
Keywords:	Optical Transport Network, Layer 1 Encryption, Wire-speed Data Throughput

Table 1: ST reference

1.2 TOE Reference

The TOE reference provides identification material for the TOE that the ST refers to.

TOE Type:	Cryptographic module that implements user data confidentiality functions
TOE Name:	Ciena 6500 Packet-Optical Platform Flex3 WaveLogic 3e OCLD Encryption Module
TOE Version:	2.01
TOE Developer:	Ciena Corporation
TOE Hardware:	Components: • Krypto Module (part number NTK53926-501, NTK53926-502) and • ASIC (part number 077-0084-017, 077-0084-028) on the Ciena 6500 Flex3 WaveLogic 3e OCLD (NTK539QS/NTK539QV variants) circuit packs
TOE Firmware:	• KM Firmware Version 2.01 • ASIC Firmware Version 1.00

Table 2: TOE reference

The difference between the QS and QV circuit packs is essentially the optical performance on the DWDM line side - the QS is rated for long haul reach (1000s of km) vs. metro reach (~300km) for the QV variant.

1.3 TOE Overview

1.3.1 Introduction

The endpoints of network device communication can be geographically and logically distant and may pass through a variety of other potentially untrusted systems. To protect any critical user data traffic the security functionality of the network device must provide appropriate cryptographic services.

Thus, the Target of Evaluation (TOE) addressed by this Security Target is a cryptographic subsystem comprising firmware and hardware implemented as components on a line card or circuit pack (the Ciena 6500 Flex3 WaveLogic 3e OCLD Circuit Pack), which can be installed with the Ciena 6500 series Packet-Optical Platform.

The TOE offers an integrated transport encryption solution providing protocol-agnostic 100 Gb/s or 200 Gb/s wire-speed encryption service.

1.3.2 TOE Type

The Target of Evaluation (TOE) is a network-related cryptographic subsystem that implements wire-speed encryption of user data transported to a peer cryptographic subsystem over the Optical Transport Network (OTN). The cryptographic security functions protect the confidentiality of user data and thus provide security services according to an organizational security policy. The TOE generates, uses, manages and protects the cryptographic keys used for these cryptographic security functions.

1.3.3 TOE Usage and Major Security Features

Typically, the TOE respectively the WaveLogic 3e OCLD circuit pack product is installed in two 6500 Packet-Optical Platform shelves which are connected via the OTN as shown in the following figure:

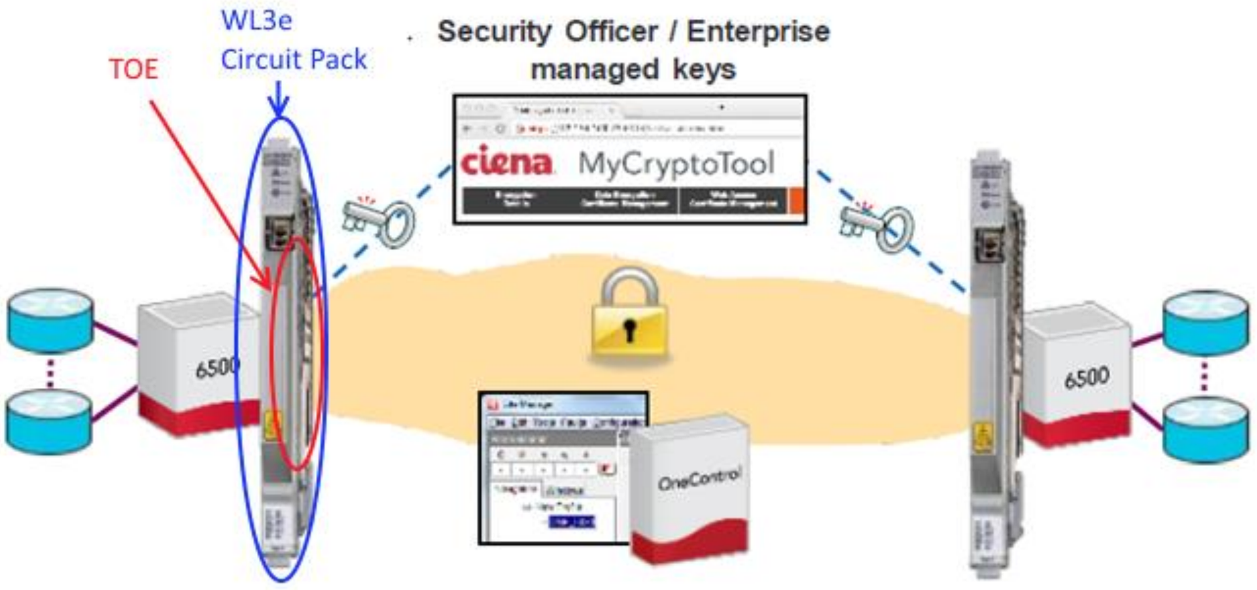


Figure 1: TOE usage scenario

In this environment the TOE provides the following major security features to guarantee the confidentiality of the user data transferred on the OTN:

- verification of integrity and authenticity of the TOE's firmware during start-up,
- mutual peer authentication between the TOE and the peer device connected via OTN,
- regular agreement on new encryption/decryption keys between the TOE and the peer,
- encryption/decryption of all user data transferred between the TOE and the peer node,

Thus, the TOE serves as a cryptographic subsystem responsible for encryption/decryption of user data exiting or entering the Flex3 WL3e OCLD circuit pack on its way to a peer device, which is connected via the OTN.

Configuration and management of the TOE security functions are done by a Crypto Officer who connects to the TOE via a TLS 1.2 mutual authenticated HTTPS connection to use the services of the MyCryptoTool application as shown in the figure above.

Chassis management and Flex3 WL3e OCLD circuit pack management for non-security-related configuration and carrier provisioning are done by an operator using the Transport Control Subsystem (TCS) which does not belong to the TOE.

The Crypto Officer management role is referred to as administrator in the following.

1.3.4 Non-TOE Hardware/Software/Firmware

The following figure outlines the main hardware components of the Ciena 6500 Packet-Optical Platform Flex3 WaveLogic 3e OCLD Encryption Module circuit pack and shows those parts of the circuit pack that are considered as non-TOE, too.

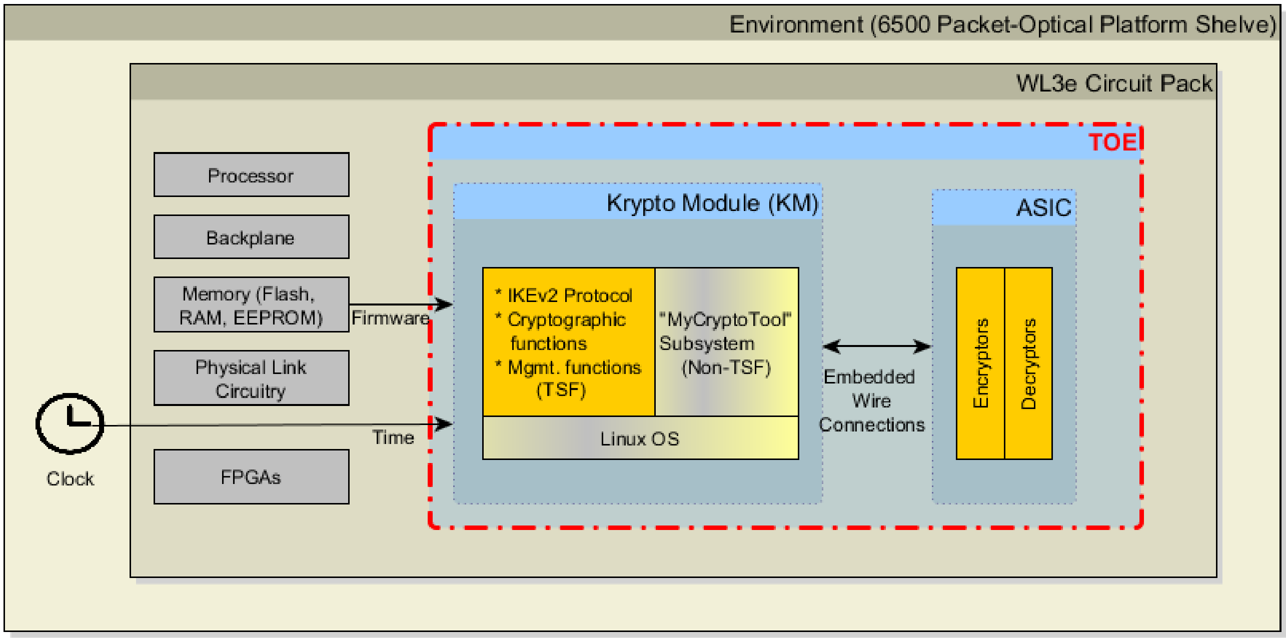


Figure 2: Non-TOE components within the Ciena 6500 Flex3 WaveLogic 3e OCLD Circuit Pack

On the Ciena 6500 Flex3 WaveLogic 3e OCLD Circuit Pack the following hardware components are considered as the TOE hardware:

- the Krypto Module (KM),
- the ASIC,
- the embedded wire connections.

All other hardware components, i.e.:

- the Processor on the circuit pack,
- the Backplane Interface,
- the attached Memory (Flash, RAM, EEPROM),
- the Physical Link Circuitry,
- the FPGAs,
- the KM module's/ASIC casing, heatsink, heatspreader, tamper-evident labels and screws
- the real time clock.

do not belong to the TOE.

The TOE software consists of an implementation of the Internet Key Exchange Protocol Version 2 (IKEv2) running on the KM with an embedded Linux OS. This IKEv2 implementation is responsible for establishing an authenticated and encrypted secure communication channel between two peer nodes for key management, i.e. an IKE Security Association (IKE SA) is established and further used to derive symmetric AES keys for datapath encryption and decryption which is in turn executed by the ASIC.

The IKEv2 protocol implementation is supported by cryptographic functions providing random number generation, hashing, ECC signature generation/verification, key derivation and symmetric protocol data encryption/decryption.

For management access to the Ciena 6500 Flex3 WaveLogic 3e OCLD Module security functions there exists a Web server software on the KM called “MyCryptoTool”. This application can be accessed by an administrator via HTTPS. It is assumed that the communication subsystem of this Web service responsible for establishing the mutually authenticated HTTPS session with the administrator is non-TSF software, as this functionality might be subject to a separate evaluation according to the collaborative Protection Profile for Network Devices (NDcPP).

Thus, only the management functions executed by the MyCryptoTool application that can be called via this interface, i.e. only the internal commands executed by the MyCryptoTool application to change/manage the TOE config parameters are part of the TSF and are subject to the TOE evaluation.

Additionally, the ASIC firmware responsible for AES en-/decryption of the user datapath belongs to the TOE.

1.4 TOE Description

1.4.1 Introduction

The TOE description explains the TOE in more technical detail than it has been provided in the TOE Overview.

The following figure shows the TOE (confined by the dotted red line) as part of the Ciena 6500 Flex3 WaveLogic 3e OCLD Circuit Pack which is installed in the 6500 Packet-Optical Platform shelf.

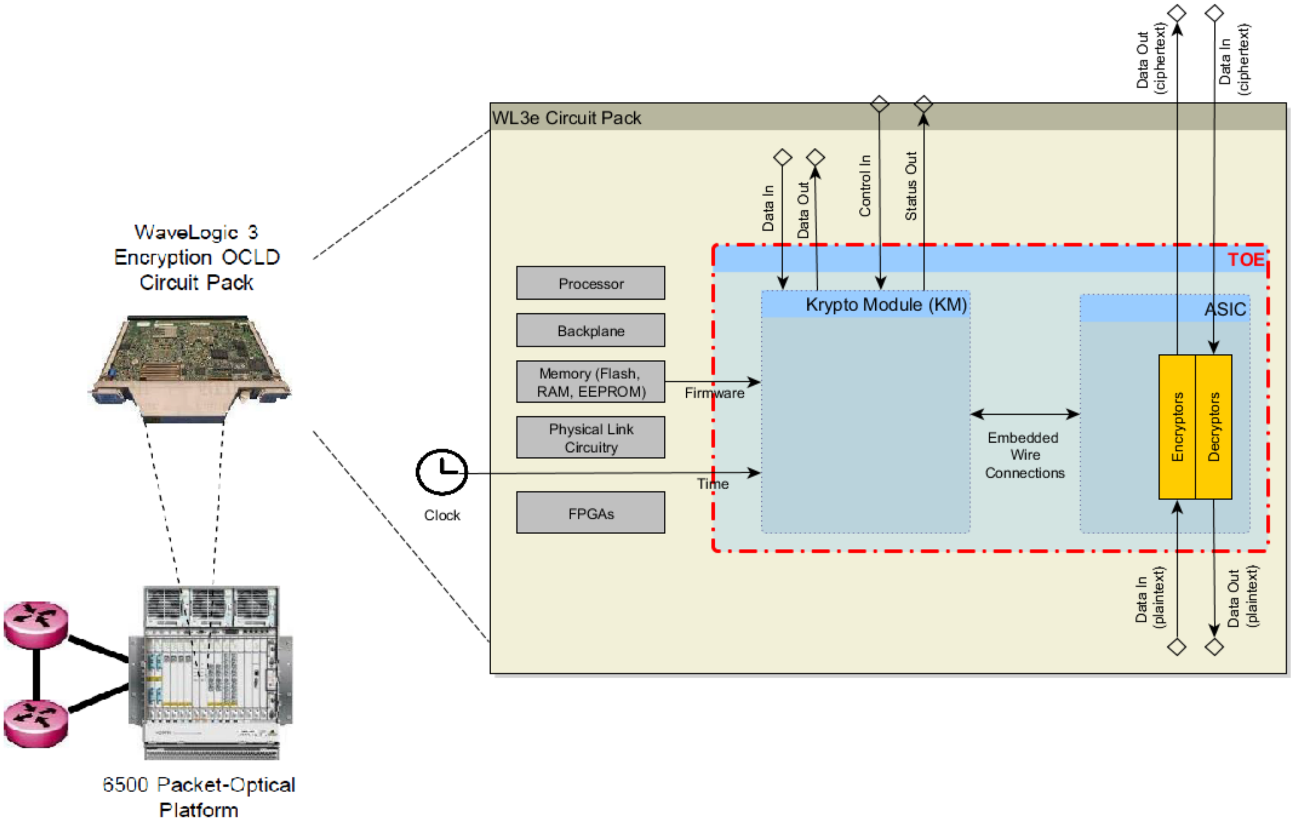


Figure 3: The TOE on the Circuit Pack plugged into the Ciena 6500 Packet-Optical Platform

1.4.2 TOE Logical Scope and Boundary

The TOE is logically defined by the implemented cryptographic algorithms and protocols, which provide the following security functions:

1. Encryption of plaintext user data entering the TOE and leaving the TOE as ciphertext data to ensure confidentiality during transfer to a peer device. The encryption key for this operation is kept confidential by the TOE.

2. Decryption of ciphertext data sent from a peer device entering the TOE and leaving the TOE as plaintext data for the user. The decryption key for this operation is kept confidential by the TOE.

3. Digital signature creation to support authentication against the peer device. The signature-creation key is kept private by the TOE.

4. Digital signature verification, which allows authentication of the peer device. The signature verification key is authentically assigned to the holder of the signature-creation key and is public available to the verifier.

The TOE manages the cryptographic keys necessary for its implemented cryptographic algorithms and protocols. The cryptographic key management controls the generation, storage, access and use of the cryptographic keys by the cryptographic functions. The cryptographic key management includes:

1. Generation of random numbers using a deterministic random number generator seeded by a physical RNG
2. Implementation of key generation algorithms depending on the intended use of the keys
3. Secure storage of private keys protecting their confidentiality
4. Key agreement protocols establishing ephemeral common secrets with external peer entities

The mutual authentication of communicating entities and the key agreement are combined to initiate and keep secure communication between trusted peer devices protecting the confidentiality of the transmitted user data.

1.4.2.1 TOE Logical Interfaces

The TOE offers its security functionality via a set of external interfaces. The following Figure 4 provides an overview of the external logical interfaces of the TOE:

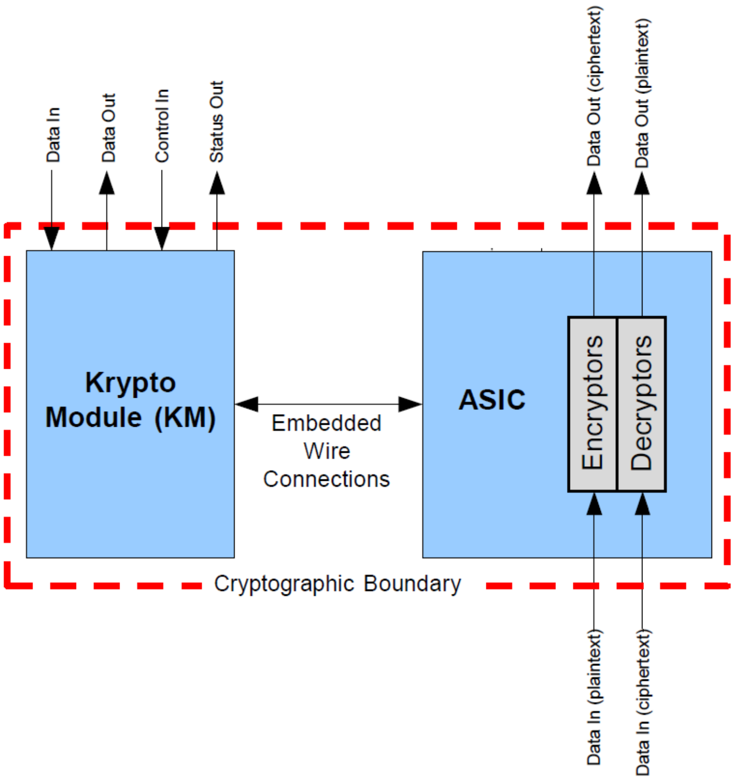


Figure 4: The TOE logical interfaces

External TOE Interfaces are provided by both the Krypto Module (KM) and the ASIC.

The KM separates its external interfaces into the following four logically distinct and isolated interface categories. They are:

- Data Input Interface

- Data Output Interface
- Control Input Interface
- Status Output Interface

Data input/output consists of the data for the IKEv2 SA services provided by the KM. Control input consists of configuration or administration data entered into the KM through the mezzanine connector of the module remotely using the MyCryptoTool interface or locally using the Transport Control Subsystem (TCS) interface. Status output consists of signals output that are then translated into alarms, LED signals, and log information by the circuit pack.

The ASIC also separates its external interfaces into the same groupings, whereas the Data Input and Output interfaces are subdivided into Backplane Data and Line Data:

- Data Input Interface
  - Backplane Data In (plaintext user data)
  - Line Data In (ciphertext)
- Data Output Interface
  - Backplane Data Out (plaintext user data)
  - Line Data Out (ciphertext)

1.4.2.2 Cryptographic Support

The following cryptographic mechanisms are implemented by the TOE:

- Random Number Generation (RNG) used for generating private ECC keys and (ephemeral) ECKA-DH keys and nonces,
- SHA-2 Hash (SHA-384),
- HMAC with SHA-2 (SHA-384),
- EC Key-Agreement Diffie-Hellman (ECKA-DH) on NIST Curve P-384 (secp384r1) as used in IKEv2,
- Pseudo-Random-Functions (PRF) as used in IKEv2 for key derivation,
- ECDSA Signature Generation/Verification (ECDSA) as used in IKEv2,
- AES-CBC with 256 bit key for private key encryption,
- AES-GCM with 256 bit key for securing the IKEv2 SA ,
- AES-CTR with 256 bit key as implemented in the ASIC for datapath encryption/decryption.

The cryptographic protocols (i.e. IKEv2) are built on top by using these cryptographic mechanisms as recommended in [TR-02102-3].

1.4.2.3 Protection of Confidentiality of User Data

The TOE protects the confidentiality of user data transferred between the TOE and connected peer devices.

Communication channels between the TOE and peer devices are secured via AES encryption in CTR mode with 256 bit keys that are agreed using IKEv2 IKE_SA_INIT and IKE_AUTH or CREATE_CHILD_SA exchanges.

1.4.2.4 Identification and Authentication

The TOE implements mechanisms that require the TOE and its peer device to be successfully identified and authenticated before allowing any user data transfer between them. There is no plaintext bypass path, i.e. traffic is either squelched or encrypted depending on successful IKEv2/ECDSA based peer authentication.

Devices are identified and authenticated through their public key X.509 certificate during an IKEv2 IKE_AUTH exchange. The corresponding private key is stored encrypted by a symmetric key encryption key within the TOE.

1.4.2.5 Management of Security Functions

The TOE implements the configuration and management of security functions as defined in this ST in FMT_SMF.1.

For this purpose the module offers two management interfaces:

- MyCryptoTool Interface – used for security-related configuration and management of the module.

  A Crypto Officer gains access to the security-related management by establishing a mutually-authenticated HTTPS/TLS session to the MyCryptoTool application using trustable X.509 certificates.

- TCS Interface – used for non-security-related configuration and carrier provisioning of the module and also firmware loads.

  An operator gains authorization over the TCS interface using a username and password credential in the form of a preshared HMAC-SHA-256 authentication string.

Application Note 1**:** For this ST it is assumed and must be ensured by organizational controls that access to both management interfaces is only possible for trustworthy personal whose identity is known and which has been authenticated.

1.4.2.6 Self-Protection

During startup the TOE verifies that the TOE’s firmware has not been compromised and is authentic. Additionally, both the KM and the ASIC run some known answer tests (KAT) to verify the correctness of the cryptographic algorithms implemented in the sub-modules.

1.4.3 TOE Physical Scope and Boundary

The physical scope of the TOE consists of:

- Hardware components
- Firmware/Software components

1.4.3.1 Hardware Components

The hardware components of the TOE appear on the circuit pack as can be seen in Figure 5:

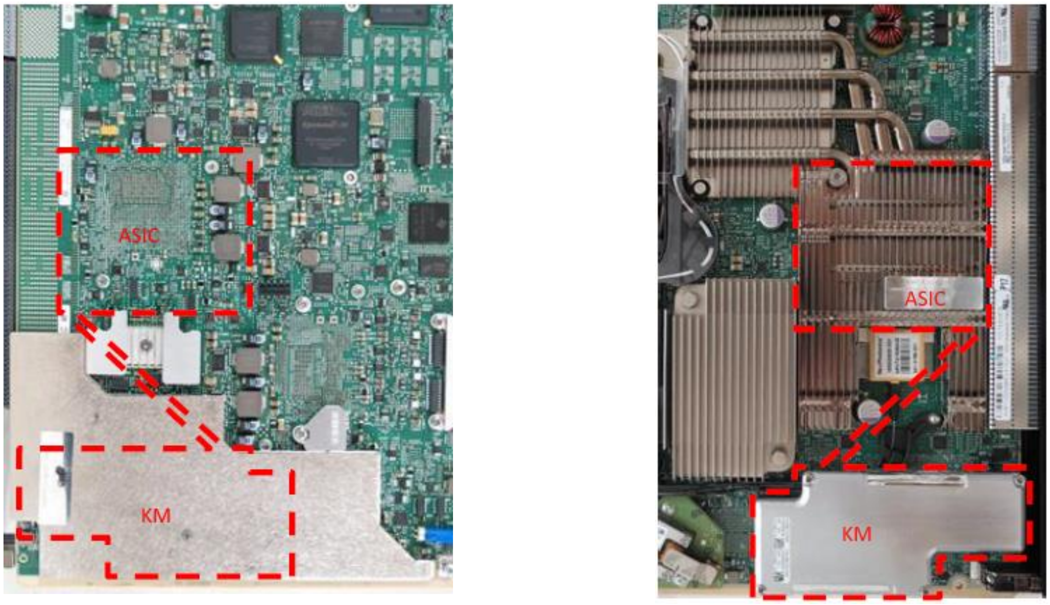


Figure 5: The TOE on the Ciena 6500 Flex3 WaveLogic 3e OCLD Circuit Pack (Bottom & Top View)

The WL3e Encryption Module is composed of a Krypto Module (KM) daughtercard, an ASIC and the PCB-embedded wire connections between them.

The KM is contained in a strong, hard metal enclosure, and is protected by two tamper-evident labels (3, 4). The wire connections between the KM and ASIC are protected from view and from tampering by multiple PCB layers. The bottom of the PCB where the KM connects is protected using the heat spreader and tamper-evident label (1). The ASIC component of the module is protected by the installed heatsink and security plate, with one tamper-evident label over one of the screws (2).

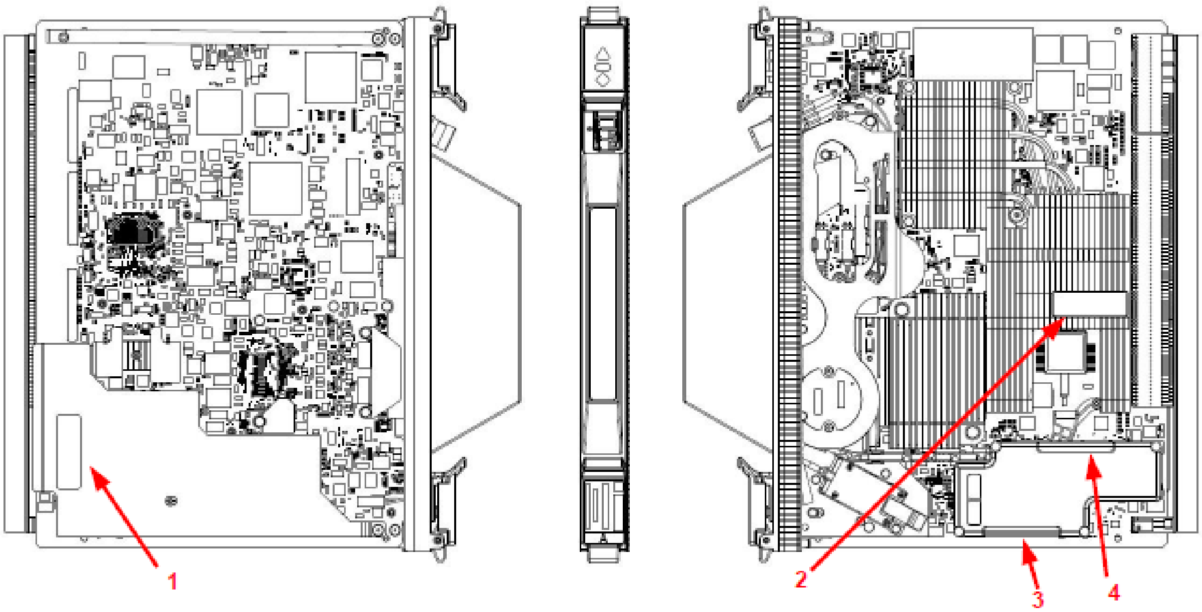


Figure 6: Tamper-Evident Label Locations

Application Note 2**:** Even though the Flex3 WaveLogic 3e OCLD Circuit Pack shows those physical security features they are not subject to the evaluation according to this ST.

The KM contains a Xilinx Zynq 7020 (Xilinx XC7Z020) with Cortex A9 dual-core processor running an embedded Linux kernel version 3.10 in a non-modifiable operational environment. The Linux operating system on the KM is not modifiable by the operator, and only the KM firmware's signed image can be executed.

The HunQ v3 ASIC contains an embedded ARM946E-S ARM processor with 128 KB of ITCM (Instruction Tightly Coupled Memory), 128 KB of DTCM (Data Tightly Coupled Memory), 8 KB of instruction cache, and 4 KB of data cache. Program and data storage is provided by 64 KB of ROM and 2 MB of RAM. The ASIC firmware is stored in the ROM prior to being loaded into RAM. While the ASIC firmware is still in ROM, a 32-bit CRC check of the ROM bootloader is performed. If successful, the firmware is loaded into RAM. Immediately upon loading into RAM, an ECDSA signature verification test using NIST P-384 curve is performed on the firmware to ensure that the image has not been modified or corrupted in any way. Once loaded, the ASIC operating environment cannot be modified.

The physical ports and interfaces of the WL3e Encryption Module consist of the KM mezzanine connector and ASIC pin-outs, as depicted in Figure 7 and Figure 8.

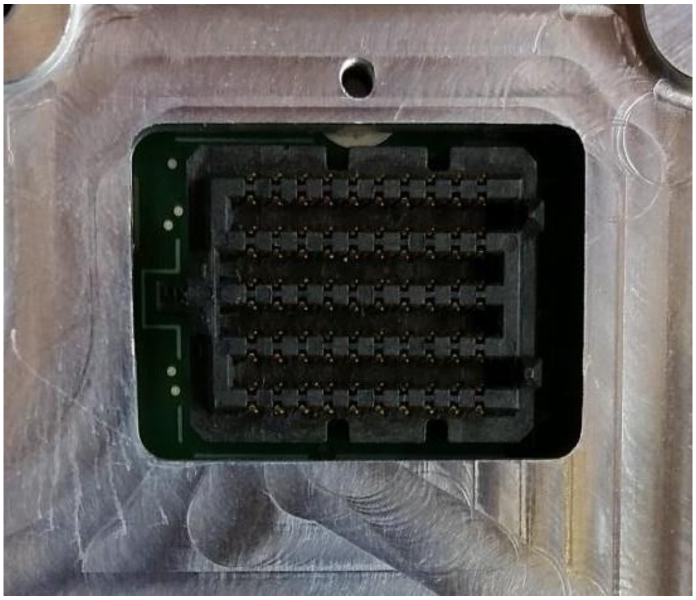


Figure 7: The KM Mezzanine connector

The KM mezzanine connector pins are grouped into the following four logically distinct and isolated interface categories:

- Data Input Interface
- Data Output Interface
- Control Input Interface
- Status Output Interface

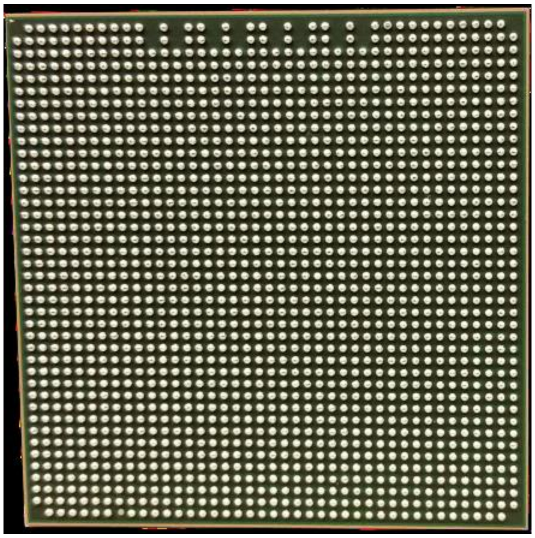


Figure 8: The ASIC Pin-Outs

The ASIC pins are categorized into the following groupings (with associated pin counts):

- Backplane Data In (40 pins)
- Backplane Data Out (40 pins)
- Line Data In (8 pins)
- Line Data Out (44 pins)
- Control In (52 pins)
- Status Out (59 pins)
- Power In (342 pins)

Remark: The ASIC also includes the following pin groupings that, based upon their purpose, are not further considered as TOE interface:

- General Purpose I/O pins: provide interfaces for pre-installation ASIC scan testing (unused/deactivated on installation)
- Internal Control/Status I/O pins: provide internal interfaces between module components
- Ground pins

1.4.3.2 Firmware/Software Components

The firmware components of the TOE consist of:

- KM Application Firmware

  The firmware is based on a hardened embedded Linux kernel version 3.10 where only the minimum number of libraries needed are contained. The Linux operating system on the KM is not modifiable by the operator, and only the KM firmware's signed image can be executed.

  The KM firmware implements the TOE's IKEv2 protocol handling, the cryptographic support and the management and configuration for that protocol.

  Application Note 3**:** The Web server subsystem within the KM firmware called "MyCryptoTool" providing an administration and management interface to the administrator is non-TSF software.

- ASIC Firmware

  The ASIC firmware implements the TOE's security functions for confidentiality of user data during OTN transfer and only the ASIC firmware's signed image can be executed.

All firmware/software components are pre-installed as binary image on the TOE hardware.

1.4.3.3 Guidance Documentation

The TOE is delivered with the following TOE guidance documents:

Name	Reference/Version
6500 Packet-Optical Platform, Release 12.3: WaveLogic Ai, Flex, 100G+, 40G, OSIC ISS, and SLIC10 Circuit Packs	323-1851-102.4 - Standard Issue 1
6500 Packet-Optical Platform, Release 12.3: Installation - General Information	323-1851-201.0 - Standard Issue 1
6500 Packet-Optical Platform, Release 12.3: Administration and Security	323-1851-301 - Standard Issue 2
6500 Packet-Optical Platform, Release 12.3: TL-1 Command Definition	323-1851-190, Standard Issue 1
6500 Packet-Optical Platform, Release 12.3: MyCryptoTool Certificate Management and Quick Start	323-1851-341 - Standard Issue 1
6500 Packet-Optical Platform, Release 12.3: Encryption and FIPS Security Policy Overview and Procedures	323-1851-340 - Standard Issue 1
Ciena 6500 Flex3 WaveLogic 3e OCLD Encryption Module: FIPS 140-2 Non-Proprietary Security Policy	Version 1.1

Table 3: TOE Guidance Documents

These documents can be accessed by Ciena customers through Ciena’s website at https://ciena.com/.

2 Conformance Claims

The conformance claims indicate the source of the collection of requirements that is met by the ST.

2.1 CC Conformance Claims

This Security Target claims conformance to

- Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and General Model; April 2017, Version 3.1, Revision 5, CCMB-2017-04-001 ([CC1]),
- Common Criteria for Information Technology Security Evaluation, Part 2: Security Functional Components; April 2017, Version 3.1, Revision 5, CCMB-2017-04-002 ([CC2]),
- Common Criteria for Information Technology Security Evaluation, Part 3: Security Assurance Requirements; April 2017, Version 3.1, Revision 5, CCMB-2017-04-003 ([CC3])

as follows:

- Common Criteria Part 2 extended (due to the use of SFR FCS_RNG.1, FDP_SDC.1 and FIA_API.1 as defined in chapter 5 of this ST)
- Common Criteria Part 3 conformant. (No extended assurance components have been defined.)

The

- Common Methodology for Information Technology Security Evaluation, Evaluation Methodology; April 2017, Version 3.1, Revision 5, CCMB-2017-04-004 ([CEM])

has to be taken into account.

2.2 PP Conformance Claims

This Security Target does not claim conformance to a Protection Profile.

2.3 Package Claims

This Security Target claims conformance to EAL2 augmented with ALC_FLR.2 as defined in CC Part 3 [CC3].

This augmentation of the predefined CC package EAL2 is denoted by the term EAL2+.

2.4 Conformance Rationale

The conformance rationale demonstrates why the chosen conformance claims were deemed appropriate.

This Security Target claims conformance to EAL2 augmented with ALC_FLR.2. This claim was chosen to ensure that the TOE has a low to moderate level of independently assured security in enforcing its security functions when instantiated in its intended environment in the absence of ready availability of the complete development record (e.g. source code).

Augmentation was chosen to demonstrate that the security measures to be followed have flaw remediation procedures in place to correct security flaws as they are reported and ensure that TOE users are aware of the corrections and the fixes.

3 Security Problem Definition

This ST describes the security problem for TOE. The intended protection of primary assets of the TOE is addressed by organizational security policies. The TOE protects the user data in confidentiality. The use of cryptographic methods implies specific threats, which are common for a TOE as a cryptographic module.

3.1 Assets

The TOE is intended to protect the following data as primary assets:

- **Plaintext user data,** i.e. user data encoded in a public known way which is transformed by an encryption algorithm into ciphertext data (i.e. plaintext as input data) or which is the result of decryption of the corresponding ciphertext data (i.e. plaintext as output data). Plaintext data contain information, which need protection in confidentiality.

- **TOE configuration**, i.e. the TOE Firmware (KM and ASIC), the current (stored) configuration data (e.g. trusted peer certificates, IKE and Child-SA Key-Exchange periods, selected TCM/GCC channel and ODU OH Reserve Encryption Bytes), which need protection in integrity.

The use of cryptographic algorithms and protocols requires the protection of the cryptographic keys as secondary assets. The cryptographic keys need protection depending on the cryptographic technique they are used for:

- **Secret keys** of symmetric and **private keys** of asymmetric cryptographic algorithms and protocols need protection in confidentiality and integrity.

- **Public keys** of asymmetric cryptographic algorithms and protocols need protection in integrity and authenticity.

The need of confidentiality of secret and private keys follows directly from the cryptographic technique. The integrity protection for these keys prevents indirect attacks (e.g. substitution of an unknown secret key by a known key compromise the subsequent encryption of plaintext data, an undetected modification of a private key may enable attacks against this key).

3.2 Threats

The threat agents, the TOE must demonstrate resistance against, are attackers with expertise, resources, and motivation that combine to have a **basic** attack potential.

The TOE addresses the following threats:

T.DisclosedUserData	An attacker may attempt to disclose data within the user data stream transmitted/received by the TOE over the untrusted OTN. If such an attack is successful, then the confidentiality of transmitted/received user data is compromised.
T.ModifiedFirmware	An attacker may attempt to infiltrate a modified firmware that the TOE would boot during start-up.

If such an attack is successful, the integrity of the whole TOE security features would be compromised.

3.3 Organizational Security Policies

The following table describes the organizational security policies (OSP) relevant to the operation of the TOE:

OSP.AuthenticatedPeer

The organizational security policy requires that user data exchange is only allowed with mutual trusted peers, i.e. both the TOE must proof its identity to the peer and the peer must be authenticated by the TOE.

OSP.TransferProtected

The organizational security policy requires that the confidentiality of data exchanged with peers is protected during transfer.

3.4 Assumptions

In this threat model the following assumptions about the environment need to be taken into account in order to ensure a secure operation of the TOE.

A.TrustedAdmin

As the security functions of the TOE can be compromised by an authorized administrator, such persons are assumed to be non-evil, well-trained and can be trusted to perform their duties correctly.

A.TrustedAdminAccess

The tools used by the administrator (e.g. Web Browser) for admin access to the TOE are assumed to be trustworthy, so that nothing evil can be caused to the TOE by their usage.

The access channels for administration are separate from the datapath channel and provide authenticity and confidentiality.

It is assumed that the administrator has been identified and authenticated prior to accessing any controlled resources of the TOE.

A.TrustedCertificates

The administrator is responsible to manage (i.e. obtain and import) the TOE’s ECC certificate necessary for identifying the TOE during IKEv2 message exchange.

It is assumed that the device certificate has been issued by a trustworthy CA.

It is assumed that the administrator imports a certificate establishing the identity of the TOE into the corresponding TOE only.

It is assumed that the certificate used by the developer to sign the TOE firmware has been issued by a trustworthy CA.

A.TrustedFactoryKeys
It is assumed that keys generated at the factory are strong random numbers and are brought into the TOE in a secure production environment.

A.ReliableTime
It is assumed that the real time clock outside of the TOE provides reliable time services for the TOE. It is assumed that this is achieved by regular time synchronization with NTP time server(s).

A.TrustedCircuitPack
It is assumed that all non-TOE parts of the Flex3 WL3e OCLD circuit pack (see Figure 2) are trustworthy and do not compromise the security functionality of the TOE.

A.PhysicalProtection
As the security functions of the TOE can be compromised by an attacker with physical access to the internetworking device containing the TOE, it is assumed that the internetworking device containing the TOE is located in a physically secure environment.

4 Security Objectives

The security objectives are a high-level and abstract statement of the intended solution to the security problem defined by the SPD. The following objectives describe how the security problem is addressed.

4.1 Security Objectives for the TOE

The following security objectives are defined for the TOE:

SOT.CryptographyForIKE

The TOE must implement the cryptographic algorithms necessary for session establishment and regular session key renewals in short intervals according to the Internet Key Exchange (IKEv2) protocol.

In addition, the TOE must implement a random number generator and shall generate the required keys, ensure that the keys are only used for an acceptable amount of time and destroy ephemeral keys if not longer needed.

SOT.DatapathEncryption

The TOE must use its ASIC module for encryption/decryption of user data payload to protect confidentiality.

SOT.ManagementForIKE

The TOE must provide the following management functions for configuring the IKE service:

- IKE SA certificate management
- IKE SA timeout parameters configuration

SOT.SelfProtection

The TOE shall implement functionality to protect its security functions against malfunctions and tampering.

Private keys for TOE identification and authentication shall be kept secret.

On startup the TOE shall:

- overwrite any information that is no longer needed to ensure that it is no longer available via the interfaces of the TOE,
- check TOE firmware for integrity and authenticity,
- perform various self-tests on the cryptographic algorithm implementations to verify their functionality and correctness.

4.2 Security Objectives for the Operational Environment

SOE.TrustedAdmin

Those responsible for the operation of the TOE must ensure that management and configuration of the security functions of the TOE is undertaken by non-evil, trusted administrators trained in the secure operation of the TOE.

SOE.TrustedAdminAccess

The tools used by the administrator (e.g. Web Browser) are non-evil and trustworthy.

Administration by an administrator is only possible through a channel which is separate from the datapath channel and which provides authenticity and confidentiality.

The administrator must be identified and authenticated prior to accessing any controlled resources of the TOE.

SOE.TrustedCertificates

The administrator is responsible to manage (i.e. obtain and enroll) the TOE's ECC certificate necessary for identifying the TOE during IKEv2 message exchange.

The ECC device certificate is signed by a trustworthy CA with a strong ECC key pair from the NIST P-384 (secp384r1) elliptic curve.

The administrator imports the certificate establishing the identity of the TOE into the proper TOE only.

The developer key pair used for signing the TOE firmware is a cryptographically strong ECC key pair from the NIST P-384 (secp384r1) elliptic curve signed by a trustworthy CA.

SOE.TrustedCircuitPack

All non-TOE parts of the Flex3 WL3e OCLD circuit pack (see Figure 2) are trustworthy and do not compromise the security functionality of the TOE.

SOE.TrustedFactoryKeys

Keys generated at the factory are strong random numbers and are brought into the TOE in a secure production environment by the manufacturer.

These keys are the symmetric 256-bit Base Key Encryption Key (BKEK) and the symmetric 256-bit Master Key Encryption Key (MKEK), see Figure 11.

SOE.ReliableTime

The real time clock in the TOE environment provides reliable time services for the TOE. This shall be achieved by regular time synchronization with NTP time server(s).

SOE.PhysicalProtection

The TOE shall be installed in a non-public environment which is physically secure. Only authorized individuals may physically access the TOE.

4.3 Security Objectives Rationale

4.3.1 Overview

The security objectives rationale shows how the security objectives correspond to assumptions, threats, and organizational security policies and provide a justification of that tracing.

The following table lists all objectives for the TOE and the operational environment to show which objectives are necessary to counter a threat, meet a policy or satisfy an assumption. The table also shows that no objective exists which does not trace back to a threat, policy, or assumption.

Security Objective Threat, Policy, Assumption	SOT.CryptographyForIKE	SOT.DatapathEncryption	SOT.ManagementForIKE	SOT.SelfProtection	SOE.TrustedAdmin	SOE.TrustedAdminAccess	SOE.TrustedCertificates	SOE.TrustedFactoryKeys	SOE.ReliableTime	SOE.TrustedCircuitPack	SOE.PhysicalProtection
T.DisclosedUserData	x	x	x								
T.ModifiedFirmware				x							
OSP.AuthenticatedPeer	x		x	x							
OSP.TransferProtected		x									
A.TrustedAdmin					x						
A.TrustedAdminAccess						x					
A.TrustedCertificates							x				
A.TrustedFactoryKeys								x			
A.ReliableTime									x		
A.TrustedCircuitPack										x	
A.PhysicalProtection											x

Table 4: Rationale for Security Objectives

4.3.2 Countering the Threats

The following sections provide more detailed information on how the threats are countered by the security objectives for the TOE and its operational environment.

4.3.2.1 T.DisclosedUserData

The threat **T.DisclosedUserData** is countered by a combination of the security objectives **SOT.CryptographyForIKE, SOT.ManagementForIKE** and **SOT.DatapathEncryption.**

SOT.CryptographyForIKE defines that the TOE will implement the cryptographic algorithms necessary for the IKEv2 protocol. The TOE uses IKEv2 to establish a secure and mutually authenticated Security Association (SA) with a peer device which serves as a secure channel to agree on encryption/decryption keys for the user datapath.

The negotiated keys are used in **SOT.DatapathEncryption** for encrypting/decrypting the user data flow. The objectives together ensure that the user data communicated between the TOE and a peer device cannot be disclosed by an attacker. The derived session keys are renewed at regular short intervals using the IKEv2 mechanisms, thus avoiding an overuse of a session key. The management of those IKEv2 timeout parameters is done via **SOT.ManagementForIKE**.

4.3.2.2 T.ModifiedFirmware

The threat **T.ModifiedFirmware** is countered by the security objective **SOT.SelfProtection**.

Through the TOE firmware integrity and authenticity checking by **SOT.SelfProtection** an attack to compromise the TOE’s security functions by altering its firmware is detected.

4.3.3 Coverage of Organizational Security Policies

The following sections provide more detailed information about how the security objectives for the environment and the TOE cover the organizational security policies.

4.3.3.1 OSP.AuthenticatedPeer

The Organizational Security Policy **OSP.AuthenticatedPeer** that mandates that the TOE and its peer are mutually identified and authenticated before data can be exchanged is covered by **SOT.CryptographyForIKE.** The IKE_AUTH exchange during the IKEv2 protocol which builds on the cryptographic operations provided by **SOT.CryptographyForIKE** addresses this security requirement.

The mutual identification and authentication during the IKE_AUTH exchange depends on device certificates, which must be enrolled by management functions from **SOT.ManagementForIKE**. Finally, the signing key belonging to the device certificate must be kept secret and protected by the TOE which is implemented by **SOT.SelfProtection**.

4.3.3.2 OSP.TransferProtected

The Organizational Security Policy **OSP.TransferProtected** that mandates that the confidentiality of data exchanged with peers is protected during transfer is directly addressed by the security objective for the TOE **SOT.DatapathEncryption**.

4.3.4 Coverage of Assumptions

The following sections provide more detailed information about how the security objectives for the environment cover the assumptions.

4.3.4.1 A.TrustedAdmin

The assumption **A.TrustedAdmin** is directly and completely covered by the corresponding security objective **SOE.TrustedAdmin.**

4.3.4.2 A.TrustedAdminAccess

The assumption **A.TrustedAdminAccess** is directly and completely covered by the corresponding security objective **SOE.TrustedAdminAccess**.

4.3.4.3 A.TrustedCertificates

The assumption **A.TrustedCertificates** is directly and completely covered by the corresponding security objective **SOE.TrustedCertificates**.

4.3.4.4 A.TrustedFactoryKeys

The assumption **A.TrustedFactoryKeys** is directly and completely covered by the corresponding security objective **SOE.TrustedFactoryKeys**.

4.3.4.5 A.ReliableTime

The assumption **A.ReliableTime** is directly and completely covered by the corresponding security objective **SOE.ReliableTime**.

4.3.4.6 A.TrustedCircuitPack

The assumption **A.TrustedCircuitPack** is directly and completely covered by the corresponding security objective **SOE.TrustedCircuitPack**.

4.3.4.7 A.PhysicalProtection

The assumption **A.PhysicalProtection** is directly and completely covered by the security objective **SOE.PhysicalProtection.**

5 Extended Components Definition

The extended components definition specifies additional functional requirements not contained in CC Part 2 [CC2] or additional assurance requirements not contained in CC Part 3 [CC3].

5.1 Extended Functional Components

5.1.1 Class FCS: Cryptographic Support

This class containing cryptographic support requirements is extended by a requirement to specify the characteristics of random number generators as defined in [KS2011] section 3.1.

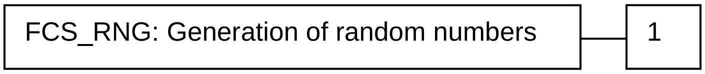
5.1.1.1 FCS_RNG Generation of random numbers

This section describes the functional requirements for the generation of random numbers, which may be used as secrets or nonces for cryptographic purposes.

Family behaviour

This family defines quality requirements for the generation of random numbers that are intended to be used for cryptographic purposes.

Component leveling



FCS_RNG.1 Generation of random numbers requires that the random number generator implements defined security capabilities and that the random numbers meet a defined quality metric.

Management: FCS_RNG.1

There are no management activities foreseen.

Audit: FCS_RNG.1

There are no actions defined to be auditable.

FCS_RNG.1	Random number generation (Class DRG.2)
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FCS_RNG.1.1	The TSF shall provide a [selection: physical, non-physical true, deterministic, hybrid physical, hybrid deterministic] random number generator that implements: [assignment: *list of security capabilities*].
FCS_RNG.1.2	The TSF shall provide random numbers that meet [assignment: *a defined quality metric*].

Table 5: Definition of Extended Component for Random Number Generation

5.1.2 Class FDP: User Data Protection

To define the security functional requirements of the TOE an additional family (FDP_SDC.1) of the Class FDP (User Data Protection) is defined here, see [SOGIS-CESGD] section 7.3.

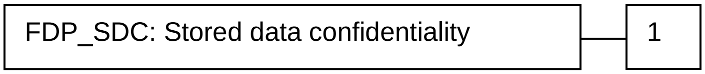
The family “Stored data confidentiality (FDP_SDC)” is specified as follows.

5.1.2.1 FDP_SDC Stored data confidentiality

Family behaviour

This family provides requirements that address protection of user data confidentiality while these data are stored within memory areas protected by the TSF. The TSF provides access to the data in the memory through the specified interfaces only and prevents compromise of their information bypassing these interfaces. It complements the family Stored data integrity (FDP_SDI) which protects the user data from integrity errors while being stored in the memory.

Component leveling



FDP_SDC.1 requires the TOE to protect the confidentiality of information of the user data in specified memory areas.

Management: FDP_SDC.1

There are no management activities foreseen.

Audit: FDP_SDC.1

There are no actions defined to be auditable.

FDP_SDC.1	Stored data confidentiality
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FDP_SDC.1.1	The TSF shall ensure the confidentiality of the information of the user data while it is stored in the [assignment: *memory area*].

Table 6: Definition of Extended Component for Stored Data Confidentiality

5.1.3 Class FIA: Identification and Authentication

The standard families of the class FIA describe only the authentication verification of users’ identity performed by the TOE and do not describe the functionality of the user to prove their identity. The following paragraph defines the extended family FIA_API from point of view of a TOE proving its identity.

5.1.3.1 FIA_API Authentication Proof of Identity

Family behaviour

This family defines functions provided by the TOE to prove its identity and to be verified by an external entity in the TOE IT environment.

Component leveling

FIA_API.1 Authentication Proof of Identity, provides proof of the identity of the TOE, an object or an authorized user or role to an external entity.

Management: FIA_API.1

The following actions could be considered for the management functions in FMT: Management of authentication information used to prove the claimed identity.

Audit: FIA_API.1

There are no actions defined to be auditable.

FIA_API.1	Authentication Proof of Identity
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FIA_API.1.1	The TSF shall provide a [assignment: *authentication mechanism*] to prove the identity of the [selection: TOE, [assignment: *object, authorized user or role*]] to an external entity.

Table 7: Definition of Extended Component for Authentication Proof of Identity

5.2 Extended Assurance Components

None specified.

6 Security Requirements

6.1 Overview

Security requirements specify the security objectives of the TOE in a standardized manner. Security requirements fall into two groups: Security Functional Requirements (SFRs) and Security Assurance Requirements (SARs).

SFRs are modeled by using components from CC Part 2 [CC2]. SARs are modeled by using components from CC Part 3 [CC3]. Necessary modifications to SFRs are performed through the permitted operations assignment, selection, iteration and refinement. This document uses the following conventions to identify the operations defined by the CC:

- **Refinement** operation (indicated by **bold text**): is used to add details to a requirement, and thus further restricts a requirement. In case that a word has been deleted from the original text this refinement is indicated by ~~**crossed out bold**~~ text.

- **Selection** operation (denoted by [underlined text enclosed in square brackets]): is used to select one or more options provided by the [CC2] in stating a requirement.

- **Assignment** operation (denoted by [*italicised text enclosed in square brackets*]): is used to assign a specific value to an unspecified parameter, such as the length of a password.

- **Iteration** operation: is identified with a suffix in the component identifier of the SFR (e.g. FCS_COP.1/IKE-SA and FCS_COP.1/Datapath).

It should be noted that the requirements in the following chapters are not necessarily be ordered alphabetically. Where useful the requirements have been grouped.

6.2 Security Functional Requirements

6.2.1 Class FCS: Cryptographic Support

6.2.1.1 Random Number generation (FCS_RNG)

FCS_RNG.1	Random number generation (Class DRG.2)
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FCS_RNG.1.1	The TSF shall provide a [deterministic] random number generator **of the pre-defined class DRG.2 according to [KS2011]** that implements: [*(DRG.2.1) If initialized with a random seed [[using the hardware-based true random number generator (TRNG) described in [Ciena-TRNG] as entropy source]], the internal state of the RNG shall [have [a min-entropy of at least 100 bits]].* *(DRG.2.2) The RNG provides forward secrecy.* *(DRG.2.3) The RNG provides backward secrecy.*].
FCS_RNG.1.2	The TSF shall provide random numbers that meet [

	(DRG.2.4) The RNG initialized with a random seed [of 384 bits at startup], generates output for which [2^{14}] strings of bit length 128 are mutually different with probability [>= ($1-2^{-8}$)][1]. *(DRG.2.5) Statistical test suites cannot practically distinguish the random numbers from output sequences of an ideal RNG. The random numbers pass test procedure A [and no additional test suites].*].

Application Note 4**:** The TSF performs all random bit generation services in accordance with NIST Special Publication 800-90A [NIST-SP800-90A] using CTR_DRBG (AES) seeded by an entropy source that accumulates entropy from one independent TSF-hardware-based noise source.

6.2.1.2 Cryptographic key operation for Hashing (FCS_COP)

FCS_COP.1/HASH	Cryptographic operation - hashing for signatures
Hierarchical to:	No other components.
Dependencies:	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction
FCS_COP.1.1/HASH	The TSF shall perform [*hashing for signature creation or verification and as basis in a Pseudo-Random Function (PRF) for IKEv2*] in accordance with a specified cryptographic algorithm [*SHA-384*] and cryptographic key sizes [*none*] that meet the following: [*NIST-FIPS-180-4, Secure Hash Standard (SHS), 2015*].

6.2.1.3 Cryptographic key management for the Key Encryption Key (FCS_CKM)

FCS_CKM.1/KEK	Cryptographic key generation – Key Encryption Key
Hierarchical to:	No other components.
Dependencies:	[FCS_CKM.2 Cryptographic key distribution, or FCS_COP.1 Cryptographic operation] FCS_CKM.4 Cryptographic key destruction
FCS_CKM.1.1/KEK	The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation **~~algorithm~~ method** [*generate a Key Encryption Key (KEK) using the Random Number Generator as specified in FCS_RNG.1*] and specified cryptographic key sizes [*256 bits*] that meet the following: [*none*].

Application Note 5: The 256-bit KEK is used for encrypting or decrypting the private key material of the IKE SA authentication certificate of the TOE which is either generated by the TOE (see FCS_CKM.1/IKE-SA-AuthCert) or

[1] The parameters chosen depend on the claimed attack potential and are taken from [KS2011] Table 13.

imported into the TOE with FDP_ITC.1/IKE-SA-AuthCert. The private key is then stored KEK encrypted (see FCS_COP.1/KEK) in non-volatile memory (see FDP_SDC.1).

6.2.1.4 Cryptographic key management for IKE SA authentication certificate (FCS_CKM)

FCS_CKM.1/IKE-SA-AuthCert	Cryptographic key generation - IKE SA authentication certificate
Hierarchical to:	No other components.
Dependencies:	[FCS_CKM.2 Cryptographic key distribution, or FCS_COP.1 Cryptographic operation] FCS_CKM.4 Cryptographic key destruction
FCS_CKM.1.1/IKE-SA-AuthCert	The TSF shall generate cryptographic **ECC** keys in accordance with a specified cryptographic key generation algorithm [*Elliptic Curve Key Pair Generation*] and specified cryptographic key sizes [*384 bits length group order*] that meet the following: [*use of EC domain parameter from FIPS186, D.1.2.4:* • *Curve P-384 (secp384r1)*].

Application Note 6: The TOE uses an IKE SA authentication certificate to identify and authenticate itself against the peer device (see FIA_API.1). The key pair used in this certificate is either generated in the TSF itself with this SFR (FCS_CKM.1/IKE-SA-AuthCert) or the keys are generated securely in the TOE's external PKI environment and then imported into the TSF during certificate import (see FDP_ITC.1/IKE-SA-AuthCert). In the first case the TSF random number generator (FCS_RNG.1) is used to generate the private key "d" in the ECC key pair (d, [d]G). The private key is then stored KEK encrypted (see FCS_COP.1/KEK) in non-volatile memory.

6.2.1.5 Cryptographic key management for IKEv2 (FCS_CKM)

FCS_CKM.1/IKEv2	Cryptographic key generation - IKEv2
Hierarchical to:	No other components.
Dependencies:	[FCS_CKM.2 Cryptographic key distribution, or FCS_COP.1 Cryptographic operation] FCS_CKM.4 Cryptographic key destruction
FCS_CKM.1.1/IKEv2	The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [*IKEv2 Session Key Generation*] and specified cryptographic key sizes [*ECDHE: 384 bits; AES: 256; HMAC: 384*] that meet the following: [*IPsec IKE v2 standards:* • *RFC-7296 sections* o *2.13 (Generating Keying Material),* o *2.14 (Generating Keying Material for the IKE SA);* o *2.17 (Generating Keying Material for Child SAs);* • *use of Diffie-Hellman group 20 from RFC-5903, section 3.2;* • *PRF-HMAC-SHA-384 from RFC-4868*].

Application Note 7**:** FCS_COP.1/HASH defines the hashing function used by the PRF-HMAC function.

Application Note 8**:** The TOE uses PRF-HMAC-SHA-384 as Pseudo-Random Function (PRF) to build a bit string for deriving AES keys for IKE SA and Child SAs according to RFC-7296. The Diffie-Hellman shared secret established during the IKE_SA_INIT or CREATE_CHILD_SA exchange is computed using the NIST P-384 (secp384r1) elliptic curve.

Application Note 9**:** The random number generator (FCS_RNG.1) is used to generate the nonces and Diffie-Hellman public values, which are transmitted in the IKE_SA_INIT and CREATE_CHILD_SA exchange.

Application Note 10**:** An IKE_SA key is renewed on regular intervals (minimum 1 hour, maximum 24 hours) using a new IKE_SA_INIT handshake. A Child-SA key is renewed on regular intervals (minimum 1 second, maximum 3600 seconds) using a new CREATE_CHILD_SA handshake.

6.2.1.6 Cryptographic key operation for Key Encryption/Decryption (FCS_COP)

FCS_COP.1/KEK	Cryptographic operation – using a KEK
Hierarchical to:	No other components.
Dependencies:	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction
FCS_COP.1.1/KEK	The TSF shall perform [*encryption and decryption of symmetric and private keys*] in accordance with a specified cryptographic algorithm [*AES-CBC*] and cryptographic key sizes [• *AES 256 bit,*] that meet the following: [• *NIST-FIPS-197 (AES)* • *AES-CBC Mode as defined in NIST-SP800-38A*].

Application Note 11**:** The TOE uses three layers of encryption keys to ultimately secure the private ECC key used by the TOE during IKEv2 authentication.

- The first key is an AES 256-bit key called Base Key Encryption Key (BKEK) which has been generated per Flex3 WL3e OCLD circuit pack and preloaded into the TOE at the factory. It is stored in plaintext in non-readable, write once, non-probable eFuse within the KM processor. During normal operation of the TOE the BKEK is used to decrypt (FCS_COP.1/KEK) the AES-256-bit key called Master Key Encryption Key (MKEK).

- The MKEK has been generated per Flex3 WL3e OCLD circuit pack at the factory and has been encrypted at the factory with the BKEK. The BKEK encrypted MKEK is stored in non-volatile memory in the KM. During normal operation of the TOE the MKEK is used to encrypt or decrypt (FCS_COP.1/KEK) the Key Encryption Key (KEK).

- The KEK is generated by the TOE (FCS_CKM.1/KEK) and is stored MKEK encrypted in non-volatile memory. During normal operation the KEK is used to encrypt or decrypt (FCS_COP.1/KEK) the private

ECC key of the device. The private ECC key of the device has been either generated (see FCS_CKM.1/IKE-SA-AuthCert) or imported (see FDP_ITC.1/IKE-SA-AuthCert) by the TOE.

6.2.1.7 Cryptographic key operation for IKE SA (FCS_COP)

FCS_COP.1/IKE-SA	Cryptographic operation – IKE SA
Hierarchical to:	No other components.
Dependencies:	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction
FCS_COP.1.1/IKE-SA	The TSF shall perform [*IKE SA data encryption, decryption, and integrity protection and mutual peer authentication by digital signature generation and digital signature verification*] in accordance with a specified cryptographic algorithm [*AES-GCM, ECDSA*] and cryptographic key sizes [• *AES 256 bit,* • *ECDSA 384 bit*] that meet the following: [• *NIST-FIPS-197 (AES)* • *AES-GCM Mode as defined in NIST-SP800-38D and RFC-5282* • *ECDSA-384 as defined in RFC-4754 on octets as specified in RFC-7296 section 2.15 according to FIPS PUB 186-4, Digital Signature Standard (DSS)], Section 6 and Appendix D, Implementing “NIST curves” P-384; [ISO/IEC 14888-3, Section 6.4, for ECDSA schemes* • *SHA-384 as defined in NIST-FIPS-180-4, Secure Hash Standard (SHS), 2015*].

Application Note 12**:** The AES 256 Bit keys for encryption and decryption of the IKE SA messages are generated with FCS_CKM.1/IKEv2. The ECDSA-384 signature generation key is the private key belonging to the TOE’s ECC X.509 certificate which has been installed in the TOE by the trusted administrator (see FDP_ITC.1/IKE-SA-AuthCert). The ECDSA-384 signature verification key is the public key in the peer’s ECC certificate which is received in the IKE_AUTH exchange.

Application Note 13**:** FCS_COP.1/HASH contains the specification for the hashing function used by the ECDSA function.

6.2.1.8 Cryptographic key operation for Datapath (FCS_COP)

FCS_COP.1/Datapath	Cryptographic operation – Datapath
Hierarchical to:	No other components.
Dependencies:	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation]

	FCS_CKM.4 Cryptographic key destruction
FCS_COP.1.1/DataPath	The TSF shall perform [*user data encryption, decryption*] in accordance with a specified cryptographic algorithm [*AES-CTR*] and cryptographic key sizes [• *AES 256 bit,*] that meet the following: [• *AES as defined in [NIST-FIPS-197]* • *CTR Mode as defined in [NIST-SP800-38A]*].

Application Note 14**:** The AES 256 Bit keys for encryption and decryption of user data are generated with FCS_CKM.1/IKEv2 during a CREATE_CHILD_SA exchange according to RFC-7296 section 2.17 (Generating Keying Material for Child SAs).

6.2.1.9 Cryptographic key operation for Firmware (FCS_COP)

FCS_COP.1/FW	Cryptographic operation – Firmware validation
Hierarchical to:	No other components.
Dependencies:	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction
FCS_COP.1.1/FW	The TSF shall perform [*digital signature verification*] in accordance with a specified cryptographic algorithm [*ECDSA*] and cryptographic key sizes [• *ECDSA 384 bit*] that meet the following: [• *ECDSA-384 with NIST-defined P-curve P-384 (secp384r1) with SHA-384 as defined in FIPS PUB 186-4, Digital Signature Standard (DSS)], Section 6 and Appendix D, Implementing “NIST curves” P-384; [ISO/IEC 14888-3, Section 6.4, for ECDSA schemes* • *SHA-384 as defined in NIST-FIPS-180-4, Secure Hash Standard (SHS), 2015*].

Application Note 15**:** The ECDSA-384 signature verification applies to the KM and ASIC firmware during start-up self-tests. The public key is programmed into the device at manufacturing time.

6.2.1.10 Cryptographic key management for key destruction (FCS_CKM)

FCS_CKM.4	Cryptographic key destruction
Hierarchical to:	No other components.
Dependencies:	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation]

FCS_CKM.4.1	The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method [• *The KEK (see FCS_CKM.1/KEK) is destroyed by zeroization via a MyCryptoTool or TCS interface command.* • *The private key belonging to the device certificate for IKA SA authentication (see FCS_CKM.1/IKE-SA-AuthCert or FDP_ITC.1/IKE-SA-AuthCert) is destroyed by zeroization of the KEK.* • *All ephemeral keys (see FCS_CKM.1/IKEv2) are destroyed by overwriting with new ephemeral key material or zeros by session termination, reboot, power removal or command via MyCryptoTool and TCS interface*] that meets the following: [*none*].

Application Note 16**:** The private key belonging to the device certificate for IKA SA authentication, which is either imported via FDP_ITC.1/IKE-SA-AuthCert or had been generated in the TOE via FCS_CKM.1/IKE-SA-AuthCert is stored encrypted with a KEK (see FCS_COP.1/KEK) in non-volatile TSF memory. Zeroization of the KEK, which encrypts all other non-ephemeral cryptographic keys (i.e. the private key of the IKE SA authentication certificate), renders the keys stored in non-volatile memory useless, thereby effectively zeroizing them. Ephemeral key material for the high speed datapath is very short lived (typically 10 seconds) and hence continuously erased/refreshed or overwritten with zeros at session termination.

6.2.2 Class FDP: User Data Protection

6.2.2.1 Import from outside of the TOE (FDP_ITC)

FDP_ITC.1/IKE-SA-AuthCert	Import of user data without security attributes – IKE SA authentication certificate with or without private key
Hierarchical to:	No other components.
Dependencies:	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] FMT_MSA.3 Static attribute initialization.
FDP_ITC.1.1/IKE-SA-AuthCert	The TSF shall enforce the [*none SFP*] when importing ~~**user data**~~ **the IKE SA authentication certificate**, controlled under the SFP, from outside of the TOE.
FDP_ITC.1.2/IKE-SA-AuthCert	The TSF shall ignore any security attributes associated with the user data when imported from outside the TOE.
FDP_ITC.1.3/IKE-SA-AuthCert	The TSF shall enforce the following rules when importing ~~**user data**~~ **the IKE SA authentication certificate** controlled under the SFP from outside the TOE: [*apply encryption as specified in FCS_COP.1/KEK to store the private key in non-volatile memory*].

Application Note 17**:** The imported device certificate for IKA SA authentication serves for identification and authentication of the TOE to peer devices, see FIA_API.1. Either the private key corresponding to the certificate has been generated within the TOE, see FCS_CKM.1/IKE-SA-AuthCert, or the private key has been generated externally and will be imported together with the data structure used in FDP_ITC.1/IKE-SA-AuthCert.

6.2.2.2 Stored data confidentiality (FDP_SDC)

FDP_SDC.1	Stored data confidentiality- private key of IKE SA authentication certificate
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FDP_SDC.1.1	The TSF shall ensure the confidentiality of the information of ~~**the user data**~~ **a private key** while it is stored in the [*non-volatile memory*].

Application Note 18**:** The private key belonging to the device certificate for IKA SA authentication, which is either imported via FDP_ITC.1/IKE-SA-AuthCert or had been generated in the TOE via FCS_CKM.1/IKE-SA-AuthCert is stored encrypted with a KEK (see FCS_COP.1/KEK) in non-volatile TSF memory.

6.2.3 Class FIA: Identification and Authentication

6.2.3.1 User identification (FIA_UID)

FIA_UID.2	**User identification before any action**
Hierarchical to:	FIA_UID.1 Timing of identification
Dependencies:	No dependencies.
FIA_UID.2.1	The TSF shall require each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.

Application Note 19**:** A peer device is identified by the TOE through its X.509 certificate submitted during the IKEv2 SA protocol handshake.

6.2.3.2 User Authentication (FIA_UAU)

FIA_UAU.2	**User authentication before any action**
Hierarchical to:	FIA_UAU.1 Timing of authentication
Dependencies:	FIA_UID.1 Timing of identification.
FIA_UAU.2.1	The TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.

Application Note 20**:** A peer device is authenticated by the TOE through validation of the submitted X.509 peer device certificate and verification of the peer device signature (see FCS_COP.1/IKE-SA) presented during the IKEv2-SA protocol handshake.

6.2.3.3 Proof of Identity (FIA_API)

FIA_API.1	Authentication Proof of Identity – IKEv2
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FIA_API.1.1	The TSF shall provide a [*authentication mechanism using ECDSA signature generation (IKEv2 RFC-7296 section 2.15 "Authentication of the IKE SA"), ECDSA with SHA-384 on the P-384 curve according to RFC-4754*] to prove the identity of the [TOE] to an external entity.

Application Note 21**:** The imported IKE SA authentication certificate (see FDP_ITC.1/IKE-SA-AuthCert) together with the corresponding private key serves for identification and authentication of the TOE to peer devices using the cryptographic operations as defined in FCS_COP.1/IKE-SA.

6.2.4 Class FMT: Security Management

6.2.4.1 Specification of Management Functions (FMT_SMF)

FMT_SMF.1	Specification of Management Functions
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FMT_SMF.1.1	The TSF shall be capable of performing the following management functions: [• *firmware load service* • *generation of a Certificate Request Message for the IKE SA authentication certificate using an asymmetric key pair generated with FCS_CKM.1/IKE-SA-AuthCert* • *import of IKE SA authentication certificate with or without private key depending on whether the key pair had been generated in the TOE or outside of the TOE* • *clearing of IKE SA authentication certificate private key* • *import/removal of trusted CA certificates* • *import/removal of trusted CA CRL* • *setting of IKEv2-parameters* o *Re-Authentication Failure Mode, determines what happens if SA re-authentication fails* o *Re-Authentication Period, i.e. IKEv2 SA re-authentication (1 hour min, 24 hours max)* o *Re-Keying Period, i.e. IKEv2 Child SA re-keying (1 second min, 3600 seconds max)*].

6.2.5 Class FPT: Protection of the TSF

6.2.5.1 Inter-TSF TSF data consistency (FPT_TDC)

FPT_TDC.1	Inter-TSF basic TSF data consistency – IKE SA authentication certificate
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FPT_TDC.1.1	The TSF shall provide the capability to consistently interpret [• *IKE SA authentication certificates,* • *CA certificates,* • *CRLs*] when shared between the TSF and another trusted IT product.
FPT_TDC.1.2	The TSF shall use [• *PKCS#12 Personal Information Exchange Syntax Standard, Version 1.1, RFC-7292 for IKE SA authentication certificates where the private key had been generated outside the TOE,* • *PEM (base64 encoded) file format for IKE SA authentication certificates where the private key had been generated within the TOE, CA certificates and CRLs*] when interpreting the TSF data from another trusted IT product.

Application Note 22**:** The TOE uses the IKE SA authentication certificate for authentication to the peer, see FIA_API.1. The CA certificate is the certificate of the CA which issues the peer device certificate. The CRL is the CRL of the CA of the peer device certificate.

6.2.5.2 TSF self test (FPT_TST)

FPT_TST.1	TSF Testing
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FPT_TST.1.1	The TSF shall run a suite of self tests [during initial start-up] to demonstrate the correct operation of [the TSF].
FPT_TST.1.2	The TSF shall provide authorised users with the capability to verify the integrity of [TSF data].
FPT_TST.1.3	The TSF shall provide authorised users with the capability to verify the integrity of [TSF].

Application Note 23**:** The TOE implements power-up self-tests checking the integrity and authenticity of the KM and the ASIC firmware (via FCS_COP.1/FW) and cryptographic algorithm tests in terms of known answer tests (KAT).

6.3 Security Assurance Requirements

The assurance requirements for the evaluation of the TOE, its development and operating environment are those from the predefined assurance package EAL2 augmented by the following components:

ALC_FLR.2 (Flaw reporting procedures)

6.4 Security Requirements Rationale

According to the two groups SFR and SAR of security requirements, the security requirements rationale is also divided into two sections: Security functional requirements rationale and security assurance requirements rationale.

6.4.1 Security Functional Requirements Rationale

6.4.1.1 Fulfilment of the Security Objectives

The following table maps security objectives to security functional requirements, showing that each security objective is covered by at least one security functional requirement and that no security functional requirement exists that is not needed by any security objective.

Security Objective of the TOE / Security Function Requirement	SOT.CryptographyForIKE	SOT.DatapathEncryption	SOT.SelfProtection	SOT.ManagementForIKE
FCS_RNG.1	x			
FCS_COP.1/HASH	x			
FCS_CKM.1/KEK			x	
FCS_CKM.1/IKE-SA-AuthCert				x
FCS_CKM.1/IKEv2	x			
FCS_COP.1/KEK			x	
FCS_COP.1/IKE-SA	x			
FCS_COP.1/Datapath		x		
FCS_COP.1/FW			x	
FCS_CKM.4	x			
FDP_ITC.1/IKE-SA-AuthCert				x
FDP_SDC.1			x	
FIA_UID.2	x			
FIA_UAU.2	x			
FIA_API.1	x			
FMT_SMF.1				x
FPT_TDC.1				x
FPT_TST.1			x	

Table 8: Rationale for Security Objectives

The inspection of the table above shows that:

- Each SFR traces back to at least one security objective,
- Each security objective for the TOE has at least one SFR tracing to it.

The following paragraphs contain more details on this mapping.

6.4.1.1.1 SOT.CryptographyForIKE

The TOE performs the cryptographic algorithms and message exchanges needed for the confidentiality and integrity of the IKE SA establishment as follows:

- **FCS_RNG.1** provides the random numbers needed for key and nonce generation in the IKEv2 protocol.
- **FCS_COP.1/HASH** is the basic cryptographic primitive used for calculating the hashes for signature generation and verification and as building block for the PRF.
- **FCS_CKM.1/IKEv2** generates the IKEv2 session keys for the IKE SA and the Child-SAs datapath encryption/decryption keys.
- **FCS_COP.1/IKE-SA** implements the cryptographic mechanisms necessary for establishment of the IKE-SA according to the Internet Key Exchange (IKEv2) protocol.
- **FCS_CKM.4** ensures that keys are destroyed in a safe way. In particular, session keys negotiated via IKE are destroyed after they are no longer needed.
- **FIA_UID.2** supports the identification of the peer device through its X.509 certificate submitted during the IKEv2 SA protocol handshake.
- **FIA_UAU.2** supports the authentication of the peer device through its X.509 certificate submitted during the IKEv2 SA protocol handshake.
- **FIA_API.1** serves for identification and authentication of the TOE to peer devices during the IKEv2 SA protocol handshake.

6.4.1.1.2 SOT.DatapathEncryption

The TOE implements the confidentiality protection of user data in the ASIC module. The following SFRs support that functionality within the TOE:

- **FCS_COP.1/Datapath** directly implements the objective SOT.DatapathEncryption.

6.4.1.1.3 SOT.SelfProtection

SOT.SelfProtection is met by a combination of the following SFRs:

- **FCS_CKM.1/KEK** generates the KEK which is used for encrypting or decrypting the private key material of the IKE SA authentication certificate and thus supports the SOT.SelfProtection objective that keys for TOE identification and authentication shall be kept secret.
- **FCS_COP.1/KEK** uses the KEK for encrypting or decrypting the private key material of the IKE SA authentication certificate.
- **FCS_COP.1/FW** supports FPT_TST.1 to verify the integrity and authenticity of the firmware and thus contributes to the TOE self protection.

- **FDP_SDC.1** implements the SOT.SelfProtection objective that keys for TOE identification and authentication shall be kept secret
- **FPT_TST.1** defines the self-testing functionality to detect whether the security functionality is working as expected.

6.4.1.1.4 SOT.ManagementForIKE

SOT.ManagementForIKE is met by a combination of the following SFRs:

- **FMT_SMF.1** lists the management functions the administrator can use to ensure the secure operation of the TOE.
- **FCS_CKM.1/IKE-SA-AuthCert** allows the generation of the cryptographic ECC key pair used in the TOE's IKE SA authentication certificate.
- **FDP_ITC.1/IKE-SA-AuthCert** allows the import of the TOE's IKE SA authentication certificate generated by a trustworthy CA.
- **FPT_TDC.1** requires the use of standardized formats for the import of certificates and private keys and thus supports FDP_ITC.1/IKE-SA-AuthCert.

6.4.1.2 Fulfilment of the SFR dependencies

The following table summarizes the existing dependencies between security requirements of this ST and demonstrates that they are fulfilled.

SFR	Dependency	Fulfilled by
FCS_RNG.1	-	-
FCS_COP.1/HASH	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction	Please refer to chapter 6.4.1.3 for missing dependency
FCS_CKM.1/KEK	[FCS_CKM.2 Cryptographic key distribution, or FCS_COP.1 Cryptographic operation] FCS_CKM.4 Cryptographic key destruction	FCS_COP.1/KEK FCS_CKM.4
FCS_CKM.1/IKE-SA-AuthCert	[FCS_CKM.2 Cryptographic key distribution, or FCS_COP.1 Cryptographic operation] FCS_CKM.4 Cryptographic key destruction	FCS_COP.1/IKE-SA FCS_CKM.4
FCS_CKM.1/IKEv2	[FCS_CKM.2 Cryptographic key distribution, or FCS_COP.1 Cryptographic operation] FCS_CKM.4 Cryptographic key destruction	FCS_COP.1/IKE-SA FCS_CKM.4
FCS_COP.1/KEK	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction	FCS_CKM.1/KEK FCS_CKM.4
FCS_COP.1/IKE-SA	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation]	FCS_CKM.1/IKE-SA-AuthCert FDP_ITC.1/IKE-SA-AuthCert

	FCS_CKM.4 Cryptographic key destruction	FCS_CKM.1/IKEv2 FCS_CKM.4
FCS_COP.1/Datapath	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction	FCS_CKM.1/IKEv2 FCS_CKM.4
FCS_COP.1/FW	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction	Please refer to chapter 6.4.1.3 for missing dependency
FCS_CKM.4	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation]	FCS_CKM.1/KEK FCS_CKM.1/IKE-SA-AuthCert FDP_ITC.1/IKE-SA-AuthCert FCS_CKM.1/IKEv2
FDP_ITC.1/IKE-SA-AuthCert	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] FMT_MSA.3 Static attribute initialisation.	Please refer to chapter 6.4.1.3 for missing dependency
FDP_SDC.1	-	-
FIA_UID.2	-	-
FIA_UAU.2	FIA_UID.1 Timing of identification	FIA_UID.2
FIA_API.1	-	-
FMT_SMF.1	-	-
FPT_TDC.1	-	-
FPT_TST.1	-	-

Table 9: Fulfillment of SFR dependencies

The inspection of the table above shows that:

- All dependencies between security functional requirements are satisfied, except as noted below in the next section

6.4.1.3 Justification of missing SFR dependencies

The hash algorithm as defined in FCS_COP.1/HASH does not need any key material. As such the dependency to an import or generation and destruction of key material is omitted for this SFR.

The signature validation as defined in FCS_COP.1/FW uses a public key that has been preloaded at the factory. The public keys are key wrapped with BKEK and written to boot flash device. The TOE does not generate or destroy this public key.

The TOE environment ensures that only trustworthy administrators can access the TOE for management (see SOE.TrustedAdmin, SOE.TrustedAdminAccess). Thus, importing the IKE SA authentication certificate FDP_ITC.1/IKE-SA-AuthCert does not require TSF access control, FDP_ACC.1 and FMT_MSA.3 are therefore not required as TOE SFRs.

6.4.2 Security Assurance Requirements Rationale

In order to keep evaluations according to this Security Target commercially feasible, EAL 2 has been chosen as assurance level, as this is the level that is commonly agreed under CCRA members.

Eventually, the augmentation by ALC_FLR.2 has been chosen to emphasize the importance of a structured process for flaw remediation at the developer's side.

6.4.2.1 Fulfillment of SAR Dependencies

The dependencies of the assurance requirements taken from EAL 2 are fulfilled automatically. The augmentation by ALC_FLR.2 does not introduce additional assurance components that are not contained in EAL 2.

6.4.2.2 Justification of missing SAR dependencies

None at this time.

7 TOE Summary Specification

7.1 Overview

This TOE Summary Specification (TSS) describes the TOE in terms of security functions and how the TOE meets the security functional requirements. The TSS explains the general technical mechanisms that the TOE uses for this purpose.

The TOE (i.e. the NTK539QS and NTK539QV variants of Flex3 WL3e OCLD circuit packs) supports wire speed point-to-point 100G and 200G encryption/decryption at the ODU4 payload layers (see Figure 10) with end-to-end encryption authentication/key exchange. The TOE provides support for the following rates and modulation formats:

- 100G QPSK modulation (NTK539QS);
- 200G 16QAM modulation (NTK539QV);

Encryption is always enabled on the encryption circuit pack. There is no option to disable encryption. The two end points of a secured datapath must be equipped with NTK539QS or NTK539QV circuit pack for peer authentication and payload encryption/decryption.

The TOE requires a valid Time of Day (TOD) in order to validate the data encryption certificates and establish traffic on the circuit pack.

The TOE is a hardware module with a multiple-chip embedded embodiment. The module consists of two primary components: a KM (processor complex) enclosed in an aluminum enclosure and a high speed datapath ASIC (HunQ) mounted on the motherboard's PCB and covered by a heatsink. These two components communicate via wire connections embedded beneath multiple PCB layers. The KM also contains integrated circuits, processors, Synchronous Dynamic Random Access Memory (SDRAM), flash memories (NOR6 and EEPROM), and FPGAs.

Figure 9 below provides another high level depiction of the TOE.

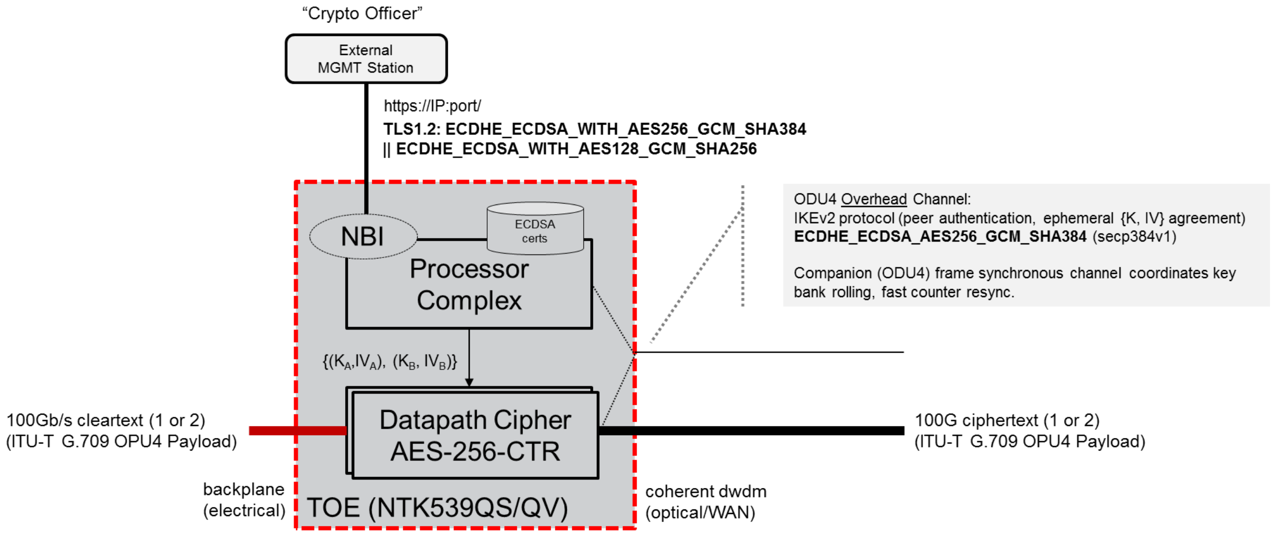

"Crypto Officer"
External MGMT Station
https://IP:port/
TLS1.2: ECDHE_ECDSA_WITH_AES256_GCM_SHA384
|| ECDHE_ECDSA_WITH_AES128_GCM_SHA256
NBI
ECDSA certs
Processor Complex
{(K_A,IV_A), (K_B, IV_B)}
Datapath Cipher AES-256-CTR
TOE (NTK539QS/QV)
100Gb/s cleartext (1 or 2)
(ITU-T G.709 OPU4 Payload)
backplane
(electrical)
coherent dwdm
(optical/WAN)
100G ciphertext (1 or 2)
(ITU-T G.709 OPU4 Payload)
ODU4 Overhead Channel:
IKEv2 protocol (peer authentication, ephemeral {K, IV} agreement)
ECDHE_ECDSA_AES256_GCM_SHA384 (secp384v1)
Companion (ODU4) frame synchronous channel coordinates key bank rolling, fast counter resync.

Figure 9: Overview of the TOE's components

The TOE components together implement the following security functions that cover the SFRs listed in Chapter 6 of this ST:

SF.FCS:	Cryptographic Support for IKEv2 and datapath encryption
SF.FDP:	User Data Protection
SF.FIA:	Identification and Authentication
SF.FMT:	Security Management
SF.FPT:	Protection of the TSF

The TSS is given in more detail in the following sections.

7.2 SF.FCS: Cryptographic Support

The TOE implements the following cryptographic mechanisms either in the high speed datapath ASIC or the KM firmware:

- Random Number Generation (RNG)
- ECC Key Generation on NIST defined P-curve P-384 (secp384r1)
- ECC Key-Agreement Diffie-Hellman (ECKA-DH) on NIST defined P-curve P-384 (secp384r1)
- ECDSA signature generation and verification for NIST defined P-curve P-384 (secp384r1)
- SHA-2 family of hash algorithms, especially
    - SHA-384
- KDF
    - PRF_HMAC_SHA2_384
- AES
    - AES-GCM
    - AES-CBC
    - AES-CTR

The implemented cryptographic mechanisms are combined and used as building blocks to finally provide high speed datapath encryption for keeping user data transferred between two connected TOE devices confidential.

The IKEv2 protocol is used to establish an authenticated and secure connection (IKEv2 Security Association) of the TOE device to its peer device. The IKEv2 SA established is used to continuously derive and establish new datapath AES keys.

ASIC Cryptographic Services: high speed (100 Gb/s) AES-256 ciphers

The primary objective of the TOE is to protect the confidentiality of the cleartext user data when transiting an untrusted OTN wide area network. This is achieved by way of high speed (100 Gb/s) AES-256-CTR ciphers embedded within the primary high speed datapath ASIC (HunQ).

The HunQ ASIC implements the AES encryption/decryption with an AES 256 Bit key in AES-CTR mode (FCS_COP.1/Datapath). The uniqueness of counter blocks across all plaintext blocks that are encrypted under a given AES 256 Bit key is guaranteed as follows:

The ICV is 128 Bit, generated from the TOE DRNG; each invocation of the AES encrypt function increments the counter by 1 mod 2^{128}; the regular {K, ICV} rotation interval is maximum 3600 seconds (see FMT_SMF.1); the ODU4 frame rate is ~856Kf/s with nominally 952 AES blocks per OTN frame. Hence the maximum number of plaintext blocks encrypted under a given {K, ICV} does not exceed 2^{42}.

The HunQ ASIC supports operation at either 200G/16QAM or 100G/QPSK depending on the modulation of the circuit pack's line interface. Hence there can actually be two fully independent AES-256-CTR ciphers within the ASIC. When configured for 200 Gb/s, both ciphers are enabled and two totally independent IKEv2 SA connections between the peers are established to generate entirely independent key material for both 100 Gb/s ciphers.

Each 100 Gb/s cleartext user data interfaces on the ASIC presents as a standard ITU-T G.709 ODU4 frame format. Each AES-256-CTR cipher operates on the OPU4 payload portion of the OTN frame. A low speed overhead channel carried within the G.709 ODU4 overhead facilitates IKEv2 peer authentication/ephemeral key agreement, hitless key bank rolling and fast counter resynchronization, see Figure 10 below.

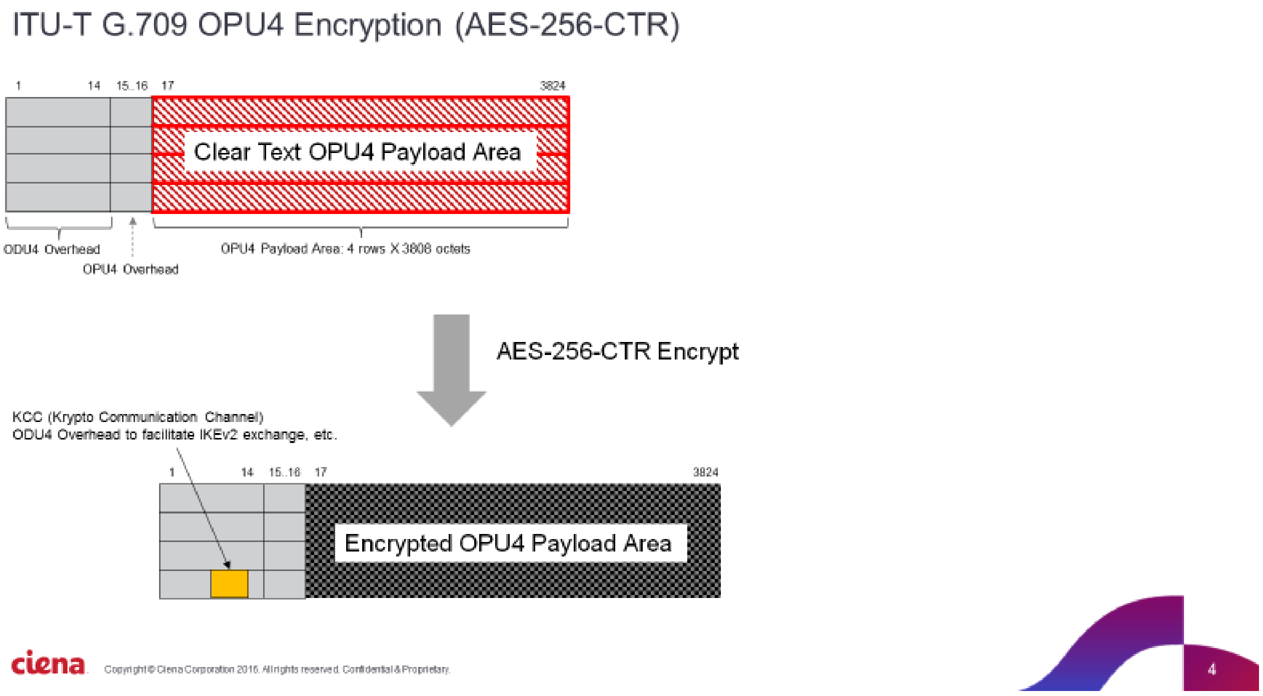

ITU-T G.709 OPU4 Encryption (AES-256-CTR)
1
14
15..16
17
3824
Clear Text OPU4 Payload Area
ODU4 Overhead
OPU4 Overhead
OPU4 Payload Area: 4 rows X 3808 octets
AES-256-CTR Encrypt
KCC (Krypto Communication Channel)
ODU4 Overhead to facilitate IKEv2 exchange, etc.
1
14
15..16
17
3824
Encrypted OPU4 Payload Area
ciena
Copyright © Ciena Corporation 2016. All rights reserved. Confidential & Proprietary.
4

Figure 10: OPU4 Encryption (AES-256-CTR)

There is no cleartext bypass path in the ASIC; the cleartext user data is either squelched (i.e. overwritten with a locally generated replacement signal, typically all 1's) or encrypted. The default behavior on power up is to squelch.

Traffic remains squelched during POST/KAT's and continues indefinitely pending successful IKEv2 peer authentication and ephemeral key agreement.

Each 100 Gb/s cipher has its own dedicated key bank. Key (K) length is 256 bits, ICV length is 128 bits. IKEv2 software configures unique {K, ICV} for each direction (i.e. onramp/encrypt {K, ICV} is distinct from offramp/decrypt {K, ICV}). Each key bank comprises current/active key material and next/standby key material to facilitate hitless key bank rolling; IKEv2 protocol agrees fresh {K, ICV} roughly once a second (maximum 3600 seconds), see FMT_SMF.1 for configuration. Periodic IKEv2 re-authentication refreshes the master session key roughly once an hour (maximum 24 hours), see also FMT_SMF.1.

The AES-256-CTR cipher of the ASIC (FCS_COP.1/Datapath) has successfully completed FIPS CAVP at NIST approved lab (AES Validation Number #4231). Upon power up, KAT's are executed to confirm cipher operation is correct and if not circuit pack is failed.

KM Cryptographic Services

The TOE's KM component implements a deterministic random number generator DRBG according to [NIST-SP800-90A, Chapter 10.2] (FCS_RNG.1). The DRBG mechanism is based on AES-256 as block cipher. The DRBG is seeded from seeding material provided by a hardware-based PTRNG, which provides an entropy source and whitening circuitry to supply a uniformly-distributed unbiased random sequence of bits to the DRBG. CRNGT (Continuous Random Number Generator Test) is implemented to ensure confidence in the entropy source. The DRBG has successfully completed FIPS CAVP at NIST approved lab (DRBG Validation Number #1315).

The TOE provides cryptographic hashing services using SHA-384 (FCS_COP.1/HASH) as specified in NIST FIPS 180-4, Secure Hash Standard (SHS), 2015. This cryptographic primitive is used in various situations, e.g. for signature generation or verification or in the PRF of IKEv2. The Hash function has successfully completed FIPS CAVP at NIST approved lab (SHS Validation Number #3469, #3468).

FCS_CKM.1/IKEv2 uses the random numbers from FCS_RNG.1 to generate an ephemeral ECC key pair in Diffie-Hellman group 20 according to RFC-5903, section "3.2 384-Bit Random ECP Group", which is used in the IKE_SA_INIT exchange as KE payload, i.e. as the Diffie-Hellman value. The shared secret computed from the Diffie-Hellman exchange and the initiator/responder nonces are then used to generate the SKEYSEED as described in RFC-7296, section 2.14. The SKEYSEED value is further used to generate all necessary IKEv2 SA keys applying the PRF function PRF-HMAC-SHA-384 iteratively as specified in RFC-7296, section 2.13. The AES 256 bit keys for datapath encryption are taken from the KEYMAT bit string as specified in RFC-7296, section 2.17.

In addition to the ephemeral ECC key generation during the IKEv2 SA establishment the TOE can generate a static ECC key pair with FCS_CKM.1/IKE-SA-AuthCert which serves as key material for the TOE's IKE SA authentication certificate. For secure storage of the corresponding private key of the certificate, the TOE stores it encrypted in non-volatile memory (see FDP_SDC.1).

Actually, the TOE uses three layers of encryption keys to ultimately secure the private ECC key used by the TOE during IKEv2 authentication.

- The first key is an AES 256-bit key called Base Key Encryption Key (BKEK) which has been generated per Flex3 WL3e OCLD circuit pack and preloaded into the TOE at the factory. It is stored in plaintext in non-readable, write once, non-probable eFuse within the KM processor. During normal operation of the TOE the BKEK is used to decrypt (FCS_COP.1/KEK) the AES-256-bit key called Master Key Encryption Key (MKEK).

- The MKEK has been generated per Flex3 WL3e OCLD circuit pack at the factory and has been encrypted at the factory with the BKEK. The BKEK encrypted MKEK is stored in non-volatile memory in the KM. During normal operation of the TOE the MKEK is used to encrypt or decrypt (FCS_COP.1/KEK) the Key Encryption Key (KEK).

- The KEK is generated by the TOE (FCS_CKM.1/KEK) and is stored MKEK encrypted in non-volatile memory. During normal operation the KEK is used to encrypt or decrypt (FCS_COP.1/KEK) the private ECC key of the device. The private ECC key of the device has been either generated (see FCS_CKM.1/IKE-SA-AuthCert) or imported (see FDP_ITC.1/IKE-SA-AuthCert) by the TOE.

The following figure shows this hierarchy of key encryption keys.

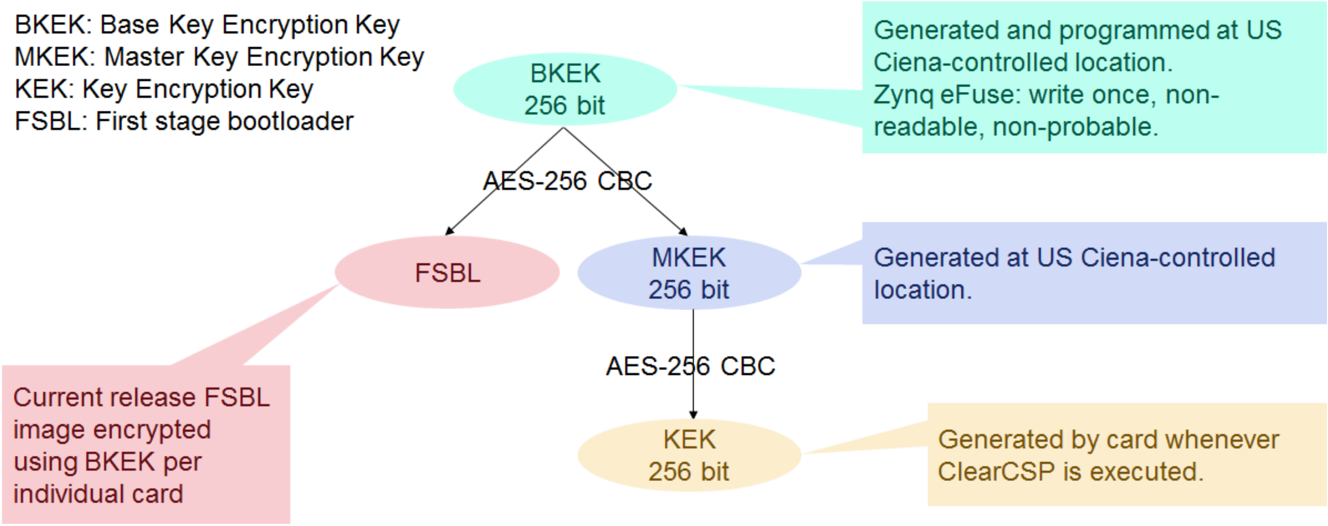

BKEK: Base Key Encryption Key
MKEK: Master Key Encryption Key
KEK: Key Encryption Key
FSBL: First stage bootloader
BKEK
256 bit
Generated and programmed at US Ciena-controlled location.
Zynq eFuse: write once, non-readable, non-probable.
AES-256 CBC
FSBL
MKEK
256 bit
Generated at US Ciena-controlled location.
AES-256 CBC
Current release FSBL image encrypted using BKEK per individual card
KEK
256 bit
Generated by card whenever ClearCSP is executed.

Figure 11: Encryption Keys Hierarchy

The SFRs FCS_CKM.1/IKEv2 and FCS_COP.1/IKE-SA are implemented by the TOE to facilitate the IKEv2 protocol message exchanges.

After FCS_CKM.1/IKEv2 helped to generate all necessary keys for the IKEv2 SA FCS_COP.1/IKE-SA supports the IKE_AUTH exchange by generating/verifying ECDSA-384 signatures for authentication purposes. The TOE's signature key is the private key belonging to its ECC X.509 certificate. This key has either been generated internally by the TOE through FCS_CKM.1/IKE-SA-AuthCert or has been installed in the TOE by the trusted administrator during the certificate enrollment process (see FDP_ITC.1/IKE-SA-AuthCert). The ECDSA-384 signature verification key is the public key in the peer's ECC certificate which is received in the IKE_AUTH exchange. The ECDSA signature generation/verification implementation has successfully completed FIPS CAVP at NIST approved lab (ECDSA Validation Number #977).

The IKE_AUTH exchange messages themselves are already encrypted and integrity protected using the AES-256-GCM cipher also implemented by FCS_COP.1/IKE-SA. The implementation of the AES-256-GCM cryptographic algorithm has successfully completed FIPS CAVP at NIST approved lab (AES Validation Number #4232).

The TOE implements the destruction of ephemeral session keys by overwriting the corresponding area in RAM with new ephemeral session keys during normal operation or with zero bytes at session termination, reboot or power removal (see FCS_CKM.4). The TOE stores all static private keys (i.e. the IKEv2 certificate private key) in non-volatile memory encrypted with the KEK. By zeroizing the KEK (FCS_CKM.4) the TOE both destroys the KEK and renders asymmetric private keys inaccessible, i.e. effectively destroying those cryptographic keys.

It is worthwhile noting that this is a “closed system”. Broad multi-vendor interoperability is intentional not required and hence the IKEv2 implementation is focused on a narrow set of options. In particular there is no need for agility in the selection of cipher suites allowing the TOE to focus on a very narrow selection of transforms (see [IKEv2-PAR]):

- Transform Type 1 - Encryption Algorithm Transform ID = 20 (ENCR_AES_GCM_16) with 256 bit AES key
- Transform Type 2 - Pseudorandom Function Transform ID = 6 (PRF_HMAC_SHA2_384)
- Transform Type 3 - Integrity Algorithm Transform ID = none (because of ENCR_AES_GCM_16 as Transform Type 1)
- Transform Type 4 - Diffie-Hellman Group Transform ID = 20 (384-bit random ECP group)
- IKEv2 Authentication Method = 10 (ECDSA with SHA-384 on the P-384 curve)

Firmware images are digitally signed (SHA-384/ECDSA-384) at time of manufacturing. Upon power-on the TOE validates this digital signature using SFR FCS_COP.1/FW executed in the FSBL, thus ensuring the TOE will only enter normal operation with a legitimate/signed Ciena software image. The ECDSA signature verification (FCS_COP.1/FW) has successfully completed FIPS CAVP at NIST approved lab (ECDSA Validation Number #977 for KM and #976 for ASIC).

7.3 SF.FDP: User Data Protection

FDP_ITC.1/IKE-SA-AuthCert allows importing the TOE’s IKE SA authentication certificate issued by a trustworthy CA in an appropriate syntax (see FPT_TDC.1). The implementation of FDP_ITC.1/IKE-SA-AuthCert ensures that the private key of the certificate imported is not stored in plaintext but encrypted with the Key Encryption Key as required by FDP_SDC.1.

The KEK itself is a 256-bit key that is stored in non-volatile memory and never exits the TOE. The KEK is encrypted with a Master Key Encryption Key (MKEK) which in turn is encrypted by a Base Key Encryption Key (BKEK) that is preloaded at the factory and stored in non-readable, write once, non-probable eFuse within the KM processor (see Figure 11).

7.4 SF.FIA: Identification and Authentication

The TOE requires each peer device to be successfully identified (FIA_UID.2) and authenticated (FIA_UAU.2) before any action can be performed.

A peer device is identified and authenticated by the TOE through validation of the submitted X.509 peer device certificate and verification of the peer device signature (see FCS_COP.1/IKE-SA) presented during the IKEv2-SA protocol handshake.

The TOE assumes the peer entity certificate as valid if all of the following conditions are met:

- peer entity certificate’s validity is checked against local system date/time

- peer entity certificate is issued by the same CA as the certificate of the TOE or one of the imported trusted CAs
- peer entity certificate serial number does not appear in the CRL for the corresponding signing CA

The administrator installs trusted CA certificate(s) and CRLs (see FMT_SMF.1) to establish roots of trust.

The other way around the TOE proofs its identity to the peer device through FIA_API.1. The TOE IKE SA authentication certificate is used to identify and the corresponding private key is used to authenticate the TOE to the peer device. Authentication is done by signing the IKE_AUTH message using the cryptographic signature generation operation as defined in FCS_COP.1/IKE-SA.

7.5 SF.FMT: Security Management

The TOE implements the security management functions as specified in FMT_SMF.1.

The management commands can be issued by the administrator using the Web based application “MyCryptoTool”. The administrator connects via TLS 1.2 (HTTPS) to the “MyCryptoTool” application. The session is mutually authenticated and secured via TLS_ECDHE_ECDSA_AES128/256_GCM_SHA384 cipher suite. Note however that the MyCryptoTool subsystem is considered as non-TSF (see Section 1.3.4).

The TOE performs a firmware update when triggered by the administrator through the TCS management interface (FMT_SMF.1). Before the TOE actually executes the firmware update the TOE verifies the integrity of the new firmware image by checking its digital signature with FCS_COP.1/FW. The corresponding public key is embedded in the active firmware code.

7.6 SF.FPT: Protection of the TSF

The TOE implements several measures to protect the TSF which are summarized in this security function.

FPT_TDC.1 requires the use of standardized encodings for the enrollment of device certificates (see FDP_ITC.1/IKE-SA-AuthCert) and the import of CA certificates and CRLs.

The following objects must be in PEM (base64 encoded) file format:

- Signing CA certificate
- Trusted CA certificates
- Certificate Revocation Lists (CRLs)
- Device certificate (if key pair has been generated in the TOE)

The following object must be in PKCS#12 file format:

- Device certificate (if key pair has been generated externally)

The TOE performs the following self-tests at power-up (FPT_TST.1) to verify the integrity of the firmware images and the correct operation of the cryptographic algorithms implemented in the module:

- Integrity and authenticity test for the KM:

    - KM application firmware image using ECDSA signature verification (FCS_COP.1/FW)
- Integrity and authenticity test for the ASIC:
    - ASIC firmware image using ECDSA signature verification (FCS_COP.1/FW)
- Cryptographic algorithm tests for all implementations of the following FIPS-approved algorithms:
    - KM
        - AES Encryption Known Answer Test (KAT)
        - AES Decryption KAT
        - SHA-256, 384, 512 KAT
        - HMAC SHA-256, 384, 512 KAT
        - SP 800-90A CTR_DRBG KAT
        - ECDSA 186-4 Signature Generation Pairwise Consistency Test (PCT)
        - ECDSA 186-4 Signature Verification PCT
    - ASIC
        - AES Encryption KAT
        - AES Decryption KAT

8 Appendices

The appendixes provide explanations for abbreviations and references to relevant standards.

8.1 Acronym Definition

A.x	Assumption x on the environment
AES	Advanced Encryption Standard
ARM	Advanced RISC Machine
ASIC	Application-Specific Integrated Circuit
BKEK	Base Key Encryption Key
BSI	Bundesamt für Sicherheit in der Informationstechnik, Federal Office for Information Security
CA	Certificate Authority
CAVP	Cryptographic Algorithm Validation Program
CBC	Cipher Block Chaining
CC	Common Criteria
CCRA	Common Criteria Recognition Arrangement
CEM	Common Evaluation Methodology for Information Technology Security
CLI	Command Line Interface
CMVP	Cryptographic Module Validation Program
CO	Crypto Officer
CRL	Certificate Revocation List
CRNGT	Continuous Random Number Generator Test
CSE	Communications Security Establishment
CSP	Critical Security Parameter
CTR	Counter
DEK	Data Encryption Key
DES	Data Encryption Standard

DH	Diffie-Hellman
DRBG	Deterministic Random Bit Generator
DTCM	Data Tightly Coupled Memory
EAL	Evaluation Assurance Level
EAL2	EAL level 2, predefined package of CC
EAL2+	EAL2 augmented by at least one SFR or SAR
ECDH	Elliptic Curve Diffie-Hellman
ECDSA	Elliptic Curve Digital Signature Algorithm
EEPROM	Electrically-Erasable Programmable Read-Only Memory
EMC	Electromagnetic Compatibility
EMI	Electromagnetic Interference
FAU	CC Functional class security AUdit
FCO	CC Functional class COmmunication
FCS	CC Functional class Cryptographic Support
FDP	CC Functional class user Data Protection
FIA	CC Functional class Identification and Authentication
FIPS	Federal Information Processing Standard
FMT	CC Functional class security ManagemenT
FPGA	Field Programmable Gate Array
FPR	CC Functional class PRivacy
FPT	CC Functional class Protection of the TSF
FSBL	First Stage Boot Loader
FTP	CC Functional class Trusted Path/channels
Gb/s	Gigabit Per Second
GbE	Gigabit Ethernet
GCC	General Communications Channel

GCM	Galois/Counter Mode
HMAC	(Keyed-) Hash Message Authentication Code
HTTPS	Hyper-Text Transport Protocol Secure
I/O	Input/Output
ICV	Initial Counter Vector
IT	Information Technology
ITCM	Instruction Tightly Coupled Memory
IV	Initialization Vector
KA	Key Agreement
KAT	Known Answer Test
KB	Kilobyte
KE	Key Exchange, in IKEv2 Diffie-Hellman value
KEK	Key Encrypting Key
KM	Krypto Module
LED	Light Emitting Diode
Mb/s	Megabits per second
MCLI	Management Command Line Interface
MKEK	Master Key Encrypting Key
MPLS	Multiprotocol Label Switching
ms	millisecond
N/A	Not Applicable
NDPP	Network Device Protection Profile
NDRNG	Non-Deterministic Random Number Generator
NIST	National Institute of Standards and Technology
NOR	Not Or
NTP	Network Time Protocol

OCLD	Optical Channel Laser and Detector
OS	Operating System
OSI	Open Systems Interconnection
OSP	Organizational Security Policy
OSP.x	OSP x enforced by environment and TOE
OTN	Optical Transport Network
OTR	Optical Transponder
PCB	Printed Circuit Board
PCT	Pairwise Consistency Test
PKCS	Public-Key Cryptography Standard
PKG	Public Key Generation
PKV	Public Key Validity
POST	Power-On Self-Test
PRF	Pseudo Random Function
PP	Protection Profile
QAM	Quadrature Amplitude Modulation
QPSK	Quadrature Phase Shift Keying
RAM	Random Access Memory
RISC	Reduced Instruction Set Computing
ROM	Read Only Memory
RSA	Rivest Shamir Adelman (encryption algorithm)
SAR	Security Assurance Requirement
SDH	Synchronous Digital Hierarchy
SDRAM	Synchronous Dynamic Random Access Memory
SFR	Security Functional Requirement
SFTP	Secure File Transfer Protocol

SHA	Secure Hash Algorithm
SHS	Secure Hash Standard
SOE.x	Security objective x for the environment
SONET	Synchronous Optical Networking
SOT.x	Security Objective x for the TOE
SP	Special Publication
SPD	Security Problem Definition
SSH	Secure Shell
ST	Security Target
T.x	Threat x
TCM	Tandem Connection Monitor
TCS	Transport Control Subsystem
TL1	Transaction Language One
TLS	Transport Layer Security
TOE	Target of Evaluation
TSF	TOE Security Functionality
TSFI	TSF Interface
TSP	TOE Security Policy
TSS	TOE Summary Specification
WL3e	WaveLogic 3 Extreme

8.2 References

CC1 Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and general model, April 2017, Version 3.1, Revision 5, CCMB-2017-04-001

CC2 Common Criteria for Information Technology Security Evaluation, Part 2: Security functional components, April 2017, Version 3.1, Revision 5, CCMB-2017-04-002

CC3 Common Criteria for Information Technology Security Evaluation, Part 3: Security assurance components, April 2017, Version 3.1, Revision 5, CCMB-2017-04-003

CEM Common Criteria for Information Technology Security Evaluation, Evaluation methodology; CCMB-2017-04-004, Version 3.1, Revision 5, April 2017

CIENA-TRNG Ciena Corporation, Ciena 6500 Packet-Optical Platform 4x10G, FIPS 140-2 Entropy Supplement

IKEv2-PAR Internet Key Exchange Version 2 (IKEv2) Parameters, https://www.iana.org/assignments/ikev2-parameters/ikev2-parameters.xhtml#ikev2-parameters-12, last visited 22.12.2017

KS2011 W. Killmann, W. Schindler, "A proposal for: Functionality classes for random number generators", Version 2.0, September 18, 2011

NIST-FIPS-180-4 NIST FIPS 180-4, Secure Hash Standard (SHS), 2015

NIST-FIPS-186-4 FIPS PUB 186-4, Digital Signature Standard (DSS), 2013

NIST-FIPS-197 NIST FIPS 197, Advanced Encryption Standard (AES), 2001

NIST-SP800-38A NIST Special Publication 800-38A, Recommendation for Block Cipher Modes of Operation: Methods and Techniques, December 2001

NIST-SP800-38D NIST Special Publication 800-38D, Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC, November 2007

NIST-SP800-90A NIST Special Publication 800-90A Revision 1, Recommendation for Random Number Generation Using Deterministic Random Bit Generators, June 2015

RFC-4754 IETF RFC 4754, D. Fu & J. Solinas, IKE and IKEv2 Authentication Using the Elliptic Curve Digital Signature Algorithm (ECDSA), January 2007

RFC-4868 IETF RFC 4868, S. Kelly & S. Frankel, Using HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512 with IPsec, May 2007

RFC-5282 IETF RFC 5282, D. Black & D. McGrew, Using Authenticated Encryption Algorithms with the Encrypted Payload of the Internet Key Exchange version 2 (IKEv2) Protocol, August 2008

RFC-5903 IETF RFC 5903, D. Fu & J. Solinas, Elliptic Curve Groups modulo a Prime (ECP Groups) for IKE and IKEv2, June 2010

RFC-7292 IETF RFC 7292, K. Moriarty, et al., PKCS #12: Personal Information Exchange Syntax v1.1, July 2014

RFC-7296 IETF RFC 7296, C. Kaufman, et al., Internet Key Exchange Protocol Version 2 (IKEv2), October 2014

SEC2 SEC 2: Recommended Elliptic Curve Domain Parameters, Certicom Research, Version 2.0, January 27, 2010

SOGIS-CESGD Joint Interpretation Library: SOG-IS Crypto Evaluation Scheme – Guidance Document, Guide to Evaluation of Cryptographic Devices – Draft -, Version 0.9.1, September 2017

TR-02102-3 Bundesamt für Sicherheit in der Informationstechnik, Technische Richtlinie BSI TR-02102-3, Kryptographische Verfahren: Empfehlungen und Schlüssellängen, Teil 3 – Verwendung von Internet Protocol Security (IPsec) und Internet Key Exchange, Version 2017-01

8.3 Revision History

Version	Date	Description
0.1	18.09.2017	Initial version after kick-off telephone conference with Ciena
0.2	22.09.2017	Work on SFRs
0.16	07.11.2017	First complete draft (except TSS)
0.18	15.11.2017	After internal review
0.19	18.12.2017	After input for Chapter 7 from MWW
0.20	03.01.2018	Rearrange Chapter 7 input, link with SFRs of Chapter 6
1.0	15.02.2018	Comments incorporated, questions answered by MWW
1.1	26.02.2018	Minor changes, corrections
1.2	19.03.2018	Updates after kick-off meeting at BSI
1.3	02.05.2018	Removed security algorithms from Section 1.4.2.2 that are not used by the TOE, added BSI certification id
1.4	29.05.2018	Added some functional description in Section 7.1
1.5	29.04.2019	Include review comments of CertLab
1.6	28.04.2020	Align document title with ST of 4x10G OTR, correct TOE version to 2.01, add Doc-ID