FIPS140-2 Non Proprietary Security Policy

ECI TR10_4EN Encryption Module

Hardware Part No: Board-Type=0x856B, Revision# D3

Firmware Revision: R6.3

System Name	Apollo
Sub System Name	IO
Module Name	TR10_4EN
Author	Milind Barve
Revision	1.6
Release Date	18-Apr-2016
Document ID	2EVYDPYSQPWJ-112-1807

http://www.ecitele.com

Contents

FIPS140-2 Non Proprietary Security Policy 1
References 3
1 Introduction 3
1.1 Document Organization 4
2 Module Overview 5
2.1 System Level Block Diagram 5
2.2 Module Specification 6
2.2.1 Main Firmware Components On Board 7
2.3 Module Ports & Interfaces 7
2.3.1 Data Ports 7
2.3.2 Debug Port 7
2.3.3 Management Ethernet 7
2.3.4 Summary of FIPS140-2 Interfaces 8
2.3.5 Physical versus Logical IO mappings 9
2.3.6 Roles of LEDS 9
2.4 Roles & Services 10
2.4.1 Roles 10
2.4.2 Services 11
2.5 Authentication Mechanism 12
2.6 Finite State Model 12
2.7 Physical Security 12
2.8 Environmental Failure Testing/ Protection 13
2.9 Operational Environment 13
2.10 Cryptographic Key Management 14
2.10.1 Approved Cryptographic Algorithms 15
2.10.2 Lifecycle of Keys 15
2.10.3 Key Zeroization 16
2.10.4 Protection of Keys 17
2.10.5 Storage of Keys 17
2.11 Self-Tests 17
2.11.1 Power-up Self Tests 18
2.11.2 Conditional Self Tests 19
2.11.3 Summary of Self Tests 19

2.12 Design Assurance 19
2.13 Mitigation of Attacks 20
3 Secure Operation 20
3.1 Module Installation 20
3.2 Initial Key Loading 21
3.3 Administrator Guidance 21
3.4 Security Officer 22
3.5 Documentation Note 22
3.6 Traceability & Identification 22
3.6.1 Hardware Identification 22
3.6.2 Firmware Traceability 23
Acronyms 24

References

ID	Title	Author	Revision	File Location
1	FIPS-140-2	US Dept. of commerce	2001-05-25	FIPS-140-2
2	FIPS-197 (AES)	US Dept. of commerce		FIPS-197 URL
3	FIPS standard archive	US Dept. of commerce		FIPS-Archive-URL
4	FIPS Annex-A (approved security functions)	US Dept. of commerce		FIPS-140-2 Annex-A
5	FIPS 140-2 Implementation Guidance	US Dept. of commerce		FIPS-150-2 IG
6	Recommendation for Block Cipher Modes of Operation	NIST, US Dept. of commerce		SP800-38D

1 Introduction

This document covers the non-proprietary security policy for ECI's TR10_4EN module. This document is prepared in the context of FIPS140-2 standard compliance of the module. TR10_4EN is a transponder module with line side encryption in ECI's Apollo product line.

FIPS140-2 is part of the FIPS (Federal Information Processing Standards) publications of the United States Federal Government. FIPS is targeted at applications dealing with non-classified but sensitive information. It serves as an important reference even across various non US security establishments

worldwide. Consequently vendors of information processing / communication equipment target FIPS compliance.

CMVP (Cryptographic Module Validation Program) validates cryptographic modules to FIPS140-2 and other cryptography based standards. CMVP is a joint effort between US (NIST) and Canadian (CSE) federal governments. TR10_4EN complies with CMVP requirements. CMVP compliance is a mandatory acceptance criterion for equipment vendors dealing with US & Canadian federal agencies.

ECI is marketing this card with its current firmware release (R6.3) that meets FIPS140-2 Level-1 compliance overall. The following table explicitly covers FIPS compliance level for each of the sections in this document.

Section Name	FIPS Compliance Level
Cryptographic Module Specification	1
Cryptographic Module Ports & Interfaces	1
Roles, Services & Authentication	1
Authentication	1
Finite State Model	1
Physical Security	1
Operational Environment	N/A
Cryptographic Key Management	1
Self-Tests	1
Design Assurance	1
Mitigation of Attacks	N/A

Table 1: Summary of FIPS Compliance Levels

1.1 Document Organization

The following document is organized into two main sections. The first one covers structural/ architectural aspects of TR10_4EN (hereby referred as 'the module' or 'the card') while the subsequent section covers security aspects during the operational mode.

ECI is submitting the following documents in addition to the current (**S**ecurity **P**olicy) as a part of FIPS certification:

1. Finite State Model
2. Vendor Evidence
3. Product Documentation
4. CAVP Certifications received for approved cryptographic algorithms in the module.

Extensive pointers to various relevant standards and literature references have been provided through the reference & acronym tables.

2 Module Overview

2.1 System Level Block Diagram

The following block diagram provides a system level overview of TR10_4EN transponder application. It covers the control plane as well as data-plane aspects of the card. We shall use the term 'IO card' or 'IO' to refer to TR10_4EN card in the rest of this document.

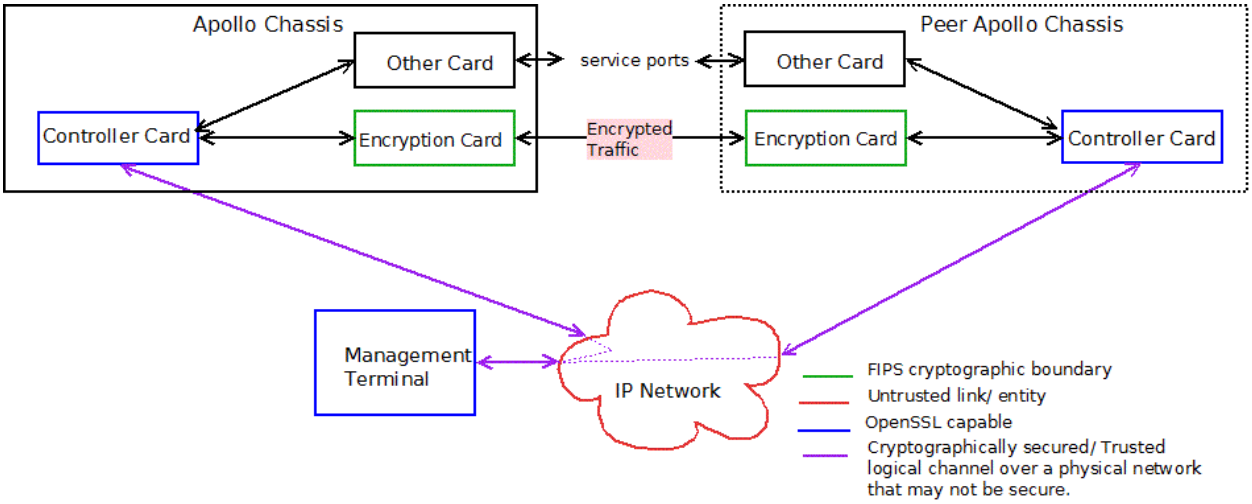

Fig 1: System Level Overview

As shown in the diagram above, the encryption card exchanges traffic in an encrypted form with remote peers. Each card has 2 line side ports; therefore it can have up to 2 remote peers. Keys for encrypting/ decrypting the traffic are established through control plane signaling.

Each line is bidirectional and there are different encryption keys active in each direction at any given time. Controller cards in each chassis are responsible for establishing these keys in a secure manner. The controller card downloads each active key to relevant IO cards in a FIPS compliant manner.

FIPS Cryptographic boundary: FIPS compliant cryptographic modules in the above diagram are marked as 'Encryption Card' with green outlines. The green outline is the cryptographic boundary for FIPS purposes.

2.2 Module Specification

TR10_4EN is a '1U' sized card that fits into ECI's Apollo chassis. ECI is seeking FIPS certification only for this module in its upcoming release (R6.3). Other card types or the Apollo chassis are not part of this FIPS certification. The module is shown in the following pictures:

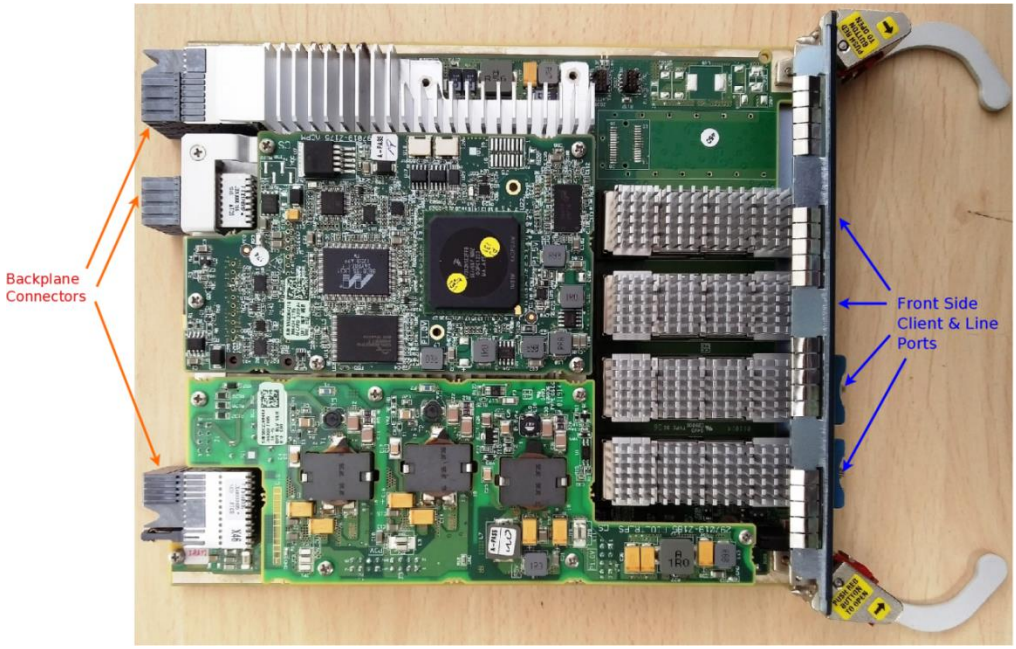

Fig. 2: TR10_4EN Physical Module (Top View)

Fig 3: TR10_4EN Physical Module (Bottom View)

The module shown in the above pictures is a 'multi-chip embedded cryptographic module' as per the FIPS terminology (FIPS140-2 section 4.5.3). It draws its power from Apollo backplane and also receives commands/ control information & parameters through its backplane Ethernet interface.

Physically the module comprises of a base board and 2 mezzanine boards. The CPU block comprising of CPU, RAM, control-FPGA and ROM (containing boot program) are on one of the mezzanine boards. The CPU is a Freescale ACPM P1012 containing a single PowerPC (e500) core. The second mezzanine board contains power-circuitry for the module.

2.2.1 Main Firmware Components On Board

1. Boot program stored in ROM, which helps load the main application image from the system card.
2. VxWorks base operational Image
3. Application binary that runs as a loadable VxWorks Module.
4. FPGA image for the Sitar FPGA.

2.3 Module Ports & Interfaces

2.3.1 Data Ports

The module has 2 pairs of client & line ports located on the front side. Being a transponder the TR10_4EN exchanges information between client port and line ports. Signal/ framing formats between a client and its corresponding line port could be different.

A picture covering the front view of the card is shown below.

Fig. 4: Ports and Interfaces on the Front Panel

The association between client and line port is fixed in this module: (C1, L1) & (C2, L2). Traffic can only flow across ports in a given pair.

2.3.2 Debug Port

The debug port is not expected to be used by the end customer. It is primarily there for ECI developers to collect debug information. It corresponds to vxWorks console port.

2.3.3 Management Ethernet

TR10_4EN is a line card in Apollo chassis that is configured & monitored through System card (sometimes called as RCP: Route Control Processor in ECI literature) in the chassis. The system card sends various image files, boot parameters as well as command and configuration parameters to each TR10_4EN module.

The management Ethernet discussed above is not accessible to end users; it's part of the backplane signals running between the system card slot(s) and each line card slot.

2.3.4 Summary of FIPS140-2 Interfaces

Interface/ Port	Interface Type	Description
Client Port C1	Data Input & Output	Located on the front panel. It exchanges data traffic with customer equipment without encryption. Traffic is cross-connected between this client port and line port L1.
Client Port C2	Data Input & Output	Located on the front panel. It exchanges data traffic with customer equipment without encryption. Traffic is cross-connected between this client port and line port L2.
Line Port L1	Data Input & Output	Located on the front panel. It exchanges OTN data traffic from remote peer always using AES256 encryption. Traffic is cross-connected between this line port and client port C1.
Line Port L2	Data Input & Output	Located on the front panel. It exchanges OTN data traffic from remote peer always using AES256 encryption. Traffic is cross-connected between this line port and client port C2.
Debug	Developer Debug	This is a small form factor serial port located on the front panel that is used by R&D and manufacturing personnel for low level debugging.
IS LED	Status output	This is a green LED on the front panel indicator showing 'in-service' status.
SW-U LED	Status output interface	This amber colored LED indicates the state of F/W health on the card
FAIL LED	Status output interface	This module colored LED indicates h/w health of the card
Power Input	Power Input	The card draws its power from the backplane.
Logical Control Channel	Control input	This is a logical control messaging channel that uses the management Ethernet physical link between the system card and TR10_4EN.
Backplane Encrypted Channel	Control Input	This is an encrypted logical channel that uses the physical Ethernet link between the system card and TR10_4EN. AES256-GCM keys for encryption/ decryption of the data traffic are input into TR10_4EN using this channel by the application running on the system card. A configuration file contains the encrypted keys. This file is transferred across the Ethernet link from system card to TR10_4EN.
Backplane Control Signals	Input signals for card control	The system card can perform certain control operations on TR10_4EN by exercising control signals such as reset and power-control

Table 2: Summary of FIPS140-2 Interfaces

All data ports are located on the front panel and have SFPs (Small Form-factor Pluggable Optical transceivers).

2.3.5 Physical versus Logical IO mappings

The following table shows relationships between Physical and logical entities that provide input-to and/ or output-from the module.

Physical Entity	Interface Type	Description of the corresponding Logical Entity
Client Ports C1, C2	Data Input & Output	A provisioned client port maps to a logical interface such as ODU or Ether-VLAN depending upon the encapsulation. Logical interfaces are used for cross-connection.
Line Ports L1, L2	Data Input & Output	A provisioned Line ports have associated 'ODU2 or ODU2e' interfaces that are used for cross-connection with their corresponding
Debug	Developer Debug	This appears as a 'tty' interface to the module firmware. It is however not used by end users.
LED(s)	Status output	All LEDS appear as register bits in the control FPGA. Physical each bit maps to a GPIO pin that's connected to an LED.
Ethernet line to system card	Control input & Status output	There is a logical control channel running on the top of TCP/IP network that physically uses the Ethernet link between TR10_4EN and active system card in the chassis.
Backplane Control Signals	Control Input	These control signals are modeled on the system card. On TR10_4EN these are not visible in f/w. These signals control CPU reset and power to the module.

Table 3: Physical vs Logical IO Mapping

2.3.6 Roles of LEDS

LEDs are 'status output' interfaces in the FIPS terminology. Roles of various LEDS on TR10_4EN front panel have been shown in the following table.

LED Name	Scope (port/ card)	Description
IS	card-level-status	The In-Service LED is green, controlled by software. It is turned on after power-up and it blinks until the software load is complete. The LED is off if the card is not provisioned by the operators. Once provisioned it is turned on.
SW/U	card-level-status	The yellow colored LED is software controlled. It shows the boot status of the card. It is only ON during the boot phase until the application firmware hasn't started executing.
FAIL	card-level-status	This is a red colored LED, which is turned on when there is a service affecting, major failure on the card.
TX	port-level-status	This is a software controlled green colored LED. It is off until a port is configured and enabled.
F/L	port-level-status	This is a red colored LED. It's off during normal operation. When on it indicates a failure condition (voltage/ temperature out-of-range, Loss-Of-Signal).

LED Name	Scope (port/ card)	Description
		It's controlled by hardware.

Table 4: LED Roles

1. The card level LEDS are switched on within 2.5 seconds after power up and then switched off after the application image loading is completed.

2. All the LEDS are switched off when the System card performs green shutdown of the card in response to a hard error.

2.4 Roles & Services

2.4.1 Roles

1. In production environments (i.e., in the field) users do not directly connect to the module (i.e., TR10_4EN) or issue commands. Only the system card sends control messages and retrieves status information from the module through its backplane Ethernet interface. We shall call this interface the control-channel.

2. The control-channel therefore effectively is a non-human or automated *crypto officer/ administrator* & *user/ security officer* of the module.

3. The control inputs from the system card to the module follow a proprietary message format. The chassis operator/ administrator with sufficient privileges on the system card provision TR10_4EN module in a given slot. This provisioning action taken at the system card leads to a set of control messages from system card to the TR10_4EN module. These messages configure the module to a point where it can pass traffic as long as it gets valid encryption keys. In the FIPS parlance, this is the role of the **crypto officer/ administrator** of the module.

4. The CSPs themselves are to be entered by **users/ security officers** as per the FIPS terminology. In the case of TR10_4EN module, AES256 keys used for encryption and decryption of traffic at each line port of the module are the only CSPs entered in the module from time to time. These CSPS are negotiated using a secure mechanism between the system card and its peers in the remote chassis. The system card however is outside the scope of the current document. However once the local system card establishes a given AES key, it is transferred to the relevant TR10_4EN modules through the control channel using FIPS compliant mechanism that will be discussed later in this document.

2.4.2 Services

The following table summarizes various services performed by the card.

Service	Role	CSP & Type of Access (R: Read, W: Write, X: Execute)	Details
Show (module) status	User	None	Input: Control message from system card to TR10_4EN module Output: reply to the above control message from TR10_4EN to the system card Interface: control channel over Ethernet
Perform Self-Test	User	X AES256-CBC (self-test of AES s/w decryption using obfuscated AES key in s/w image)	Input: The system card initiates reset of TR10_4EN module in a given slot. Output: Hardware reset is applied to various module components. Then the module runs various self-tests before initializing the module to operational state.
LED status	User	None	The LEDS mounted on the front panel of the module are output interfaces that provide visual status information about the card.
AES Key Decryption	Crypto Officer	X (AES256CBC), W	AES256-GCM keys received from the System card are in the encrypted format. The AES256-CBC implementation in the firmware decrypts these keys so that the decrypted AES256-GCM keys can be programmed in the Sitar FPGA.
Datapath Encryption	User (FPGA)	X (AES256-GCM)	ODU payload exiting each line port is encrypted using the AES256-GCM algorithm implemented in Sitar.
Datapath Decryption	User (FPGA)	X (AES256-GCM)	ODU payload received on a line port is decrypted using the AES256-GCM algorithm implemented in Sitar. The decrypted payload is then forwarded to the corresponding client port.
Initialize Module	Crypto Officer	None (No CSP is required during the initialization phase)	As part of card initialization, the ROM BOOT loads application image in RAM. The application image then initializes itself and the FPGA.

Service	Role	CSP & Type of Access (R: Read, W: Write, X: Execute)	Details
Zeroization	User	AES256-GCM (keys used by the FPGA for data path encryption decryption.) AES256-CBC (key used to encrypt/ decrypt AES256-GCM keys during backplane transfer) W	Users can zeroize the AES256-GCM & AES256-CBC keys by performing reset of the TR10_4EN module. This can be done from the System card by issuing a simple command ('request chassis power reset <slot-id>'). When the reset is issued the Sitar FPGA is reset which wipes out keys and FPGA program from it. The system RAM is also cleared and initialized by the Boot-ROM. Zeroization can be performed by any user with access privileges to the System card CLI.

Table 5: Summary of Services Provided by TR10_4EN

In each of the above service scenarios, the system card either sends one or more control messages or asserts backplane control (reset) or power lines. The system card in all these interactions implicitly plays crypto officer/ administrator and also the user/security officer roles. The backplane communication from system card to the module is assumed to be trusted. This communication is only possible if the operator can authenticate itself to the system card first. Authentication on the system card however is outside the scope of this document.

In each of the above service scenarios, encryption keys already entered into the module (if any) are not revealed under any circumstances. Moreover as a part of self-test all encryption keys (if any) previously programmed into the hardware (i.e., 'Sitar' FPGA located on the module) are zeroed.

2.5 Authentication Mechanism

The module doesn't support authentication, instead various roles are implicitly assumed by the System card.

As discussed earlier, users do not directly log on to the module to perform any administrative/ maintenance or service related operations. The System card sends proprietary commands over an inter-card Ethernet network to control the module. There is no other role based authentication involved. This model is compliant with FIPS140-2 level-1 (ref. "SECURITY LEVEL 1" requirements under section 4.3.3 from FIPS140-2).

2.6 Finite State Model

Detailed FIPS related Finite State Model (FSM) of TR10_4EN has been covered in a separate document which is part of the documentation ECI has submitted as a part of the FIPS process.

2.7 Physical Security

TR10_4EN is a line card in ECI's Apollo chassis. It falls under the category or '*multi-chip embedded cryptographic modules*'. It's FIPS140-2 level-1 compliant. From level-1 compliance perspective the card:

1. Is made of production grade components assembled mechanically on to a platter of '1U' size and has a well-defined module boundary. The platter is composed of multiple PCBS and connectors that are interlocked together through suitable mechanical means such as connectors and screws.

2. The module is inserted into the Apollo chassis and has mechanical arrangements (seen in the pictures provided in this document earlier) to fasten the card into the chassis.

3. Since it is part of a larger chassis (embedded module) it doesn't have a separate enclosure of its own. The chassis along with the front panel of the module provide a production grade cover/ enclosure to the module.

A Picture of the card seated inside an Apollo chassis is shown below:

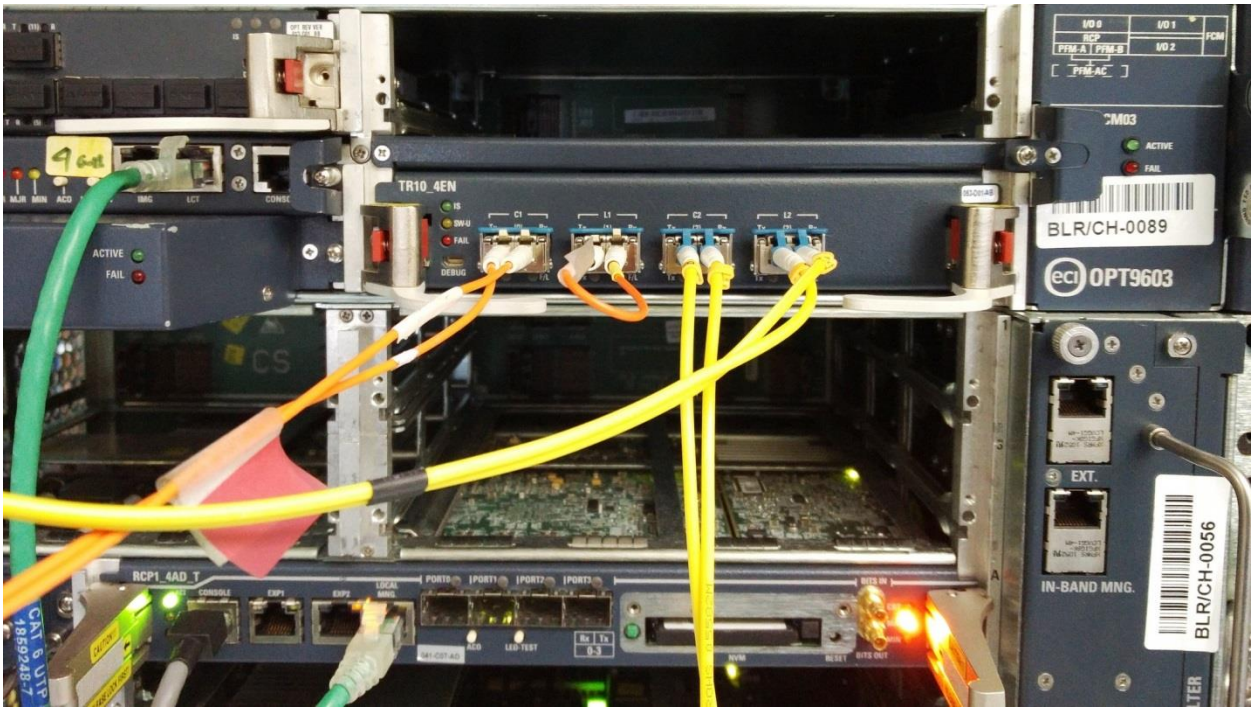

Fig. 5: TR10_4EN module In Apollo chassis

2.8 Environmental Failure Testing/ Protection

This clause (section 4.5.5 from FIPS140-2) is not applicable to the module since we are only claiming level-1 compliance.

2.9 Operational Environment

TR10_4EN module runs on VxWorks operating system which is typically used for embedded, real-time applications. Operators in the field do not have the ability to add/ delete/ modify applications running on the module. There is a single operational firmware image that is released by ECI which the System card loads on the card. Therefore the operating environment on the card is a '*limited operational environment*'. Consequently clauses in section 4.6.1 of FIPS140-2 do not apply.

VxWorks is not a multi-user operating system. The IO application runs as set of co-operating threads. The integrity check applied to the IO application image provides a basic protection against malicious code from entering and running on the system. Once the application is initialized, there is no provision to incrementally add general purpose code in the intended operating environment of the card. End users are not provided any kind of login access to the module.

The OS supports hardware interrupts. However specific interrupts have to be explicitly enabled by the application firmware. Spurious/ unexpected interrupts from any hardware source are not entertained. Interrupts facilitate real-time response from the module. The timer interrupt facilitates task/ thread switching. In general the module firmware provides a controlled environment that doesn't allow multiple users. The module does not run general purpose OS where third party programs can be loaded by either gaining access to the module or by injecting malicious (e.g., 'Trojans') code in the firmware application. Firmware integrity check is expected to catch tainted or corrupt firmware images. Integrity check failures lead to card-shutdown. As a result the operating environment provides a high degree of security.

Upon card startup, the Boot-ROM code runs from the boot-flash. This code is not field-modifiable. By definition it is '**R**ead-**O**nly-**M**emory', the integrity of which is guaranteed as long as unauthorized physical access to the chassis is blocked.

2.10 Cryptographic Key Management

There are 3 kinds of keys the module deals with.

1. Each provisioned line port needs an active encryption key to encrypt the outgoing OTN traffic. There are 2 line ports on the module. We shall call this HEK: Hardware Encryption key.

2. Each provisioned line port needs an active decryption key to decrypt the incoming traffic that is encrypted by the remote transmitting end. We shall call this HDK: Hardware Decryption Key. There are 2 line ports on the module.

3. There is an AES256 key (FAK: Firmware AES Key) embedded into the f/w operational image that is used to decrypt the above two keys that are transferred by System card to TR10_4EN module. The FAK doesn't change during the lifetime of the card. It is pre-shared between the System card and TR10_4EN module. This key is used to run AES256-CBC cipher such that the System card encrypts HEK & HDK using the FAK and the TR10_4EN module uses FAK to retrieve HEK & HDK. The module uses the AES256-CBC (#3552) implementation to decrypt (unwrap) the HEK & HDK using the FAK. The strength of the AES key wrapping is 256-bits as the FAK key is 256-bits. The AES key wrapping functionality has not been tested and is not compliant to "SP 800-38F Recommendation for Block Cipher Modes of Operation - Methods for Key Wrapping". This is a non-approved but allowed security function.

2.10.1 Approved Cryptographic Algorithms

Algorithm	Modes & Key Sizes	Function Performed	Validation Number
AES256-CBC	CBC mode. Only 256 bit key size is used	Decryption	3552
AES256-GCM	GCM mode. Only 256 bit key size is used	Encryption & Decryption	3576
AES256-ECB	(ECB) is a pre-requisite for GCM	N/A (ECB is a pre-requisite for GCM)	3551

Table 6: Approved Algorithms on TR10_4EN

2.10.2 Lifecycle of Keys

The following subsections cover usage and lifecycle of various encryption keys (which are CSPs) on the module. One thing to note is that the card never reveals any CSP in plain-text form through any of its output ports or in the firmware logs.

2.10.2.1 Hardware Encryption Key (HEK)

Each provisioned line port of TR10_4EN module needs a valid HEK in order to encrypt the ODU2 frame before transmitting. HEK is a 256-bit long entity that is used by the 'Sitar' FPGA on TR10_4EN to encrypt outgoing ODU2 frame using AES256-GCM algorithm.

HEK is established between the 2 peer chassis that have one or more line ports interconnected. All the line ports connecting a given pair of chassis share HEK to encrypt traffic in a single direction (one chassis to its peer).

HEK is **established** using a secure key setup technique (based on Diffie-Hellman algorithm) by the System card after exchanging some parameters with the System card from the peer chassis. Details related to the establishment of HEK are beyond the scope of this document. However once established, the System card needs to transfer HEK to one or more TR10_4EN modules in the chassis. This is done automatically, without operator intervention. In order to comply with FIPS140-2 requirements, the HEK is encrypted while getting transferred from System card to TR10_4EN.

HEK is ephemeral. For security reasons, a new HEK is **re-established** between a given pair of chassis periodically. The default period is 1 hour; however this parameter is configurable on the System card. The process of periodically changing the HEK is called **key-rotation**.

Upon receiving the encrypted HEK the TR10_4EN module firmware decrypts it and then **programs** the decrypted value into the FPGA. The module firmware then erases the decrypted value from RAM for security reasons.

An active HEK is replaced when a new HEK arrives into the FPGA. There is a hardware based key transition mechanism to ensure smooth HEK transitioning without traffic hits.

We have thus covered the entire lifecycle of HEK (establishment, transfer, FPGA usage, key-rotation & zeroization) in this subsection.

2.10.2.2 Hardware Decryption Key (HDK)

HDK is used to decrypt the incoming traffic at each provisioned line port. It is also a 256 bit long key used for AES-GCM decryption.

The lifecycle of HDK is similar to HEK explained above. HDK is also established by the System card (through peer messaging) and then transferred to TR10_4EN. Lifecycles of HDK and HEK are separate. Establishment and rotation of HEK and HDK occurs independent of each other.

2.10.2.3 Firmware AES Key (FAK)

FAK is a pre-shared key between System card firmware and the firmware application running on TR10_4EN. FAK does not change during the lifetime of firmware operational image of TR10_4EN.

FAK is used for one way transfer of HEK & HDK (from System card to TR10_4EN) discussed above:

1. Before transmitting an HEK or HDK, the System card encrypts these keys using FAK, a randomly generated IV and **AES256-CBC** (AES in CBC mode with 256bit key) algorithm. The IV generation mechanism meets the uniqueness requirements of FIPS140-2 IG.A-5 & SP800-38D.

2. The same algorithm (AES256 in CBC mode), IV and key (FAK) are used by TR10_4EN to decrypt the HEK, HDK received from the system card. TR10_4EN receives encrypted keys along with IV from the System card.

FAK being an AES key is technically a CSP. It is stored in an obfuscated format and split in the memory layout. FAK is assembled only before its use on the module and the assembled copy is promptly zeroized.

2.10.3 Key Zeroization

1. When self-tests are invoked on a TR10_4EN module, it gets reset. This fills the module's RAM with a standard pattern and re-initialization of FPGA which leads to zeroization of key registers inside the FPGA. Therefore triggering self-tests leads to key zeorization.

2. CSPS are never stored in plaintext in the RAM. Keys are either AES encrypted (HEK or HDK) or obfuscated (FAK). So the real zeroization is only needed for the keys (HEK, HDK) are that loaded into the FPGA.

3. Traffic encryption keys (HDK or HEK) are never stored in the plaintext form in the RAM. These keys are stored in an encrypted form in the RAM. Therefore these do not require zeroization.

4. The FPGA itself destroys any encryption keys upon getting powered down or reset. This process is near instantaneous from the time the System card initiates the zeroization procedure.

5. Any firmware maintenance (i.e., upgrade) leads to restarting the firmware and hardware on the card leading to zeroization of keys from previous incarnation.

6. The module must be removed from its chassis for any hardware maintenance. When this happens, the encryption keys in the FPGA are lost.

7. The above clauses meet level-1 FIPS compliance as described in section 4.5.1 of FIPS140-2. For level-1, zeroization triggered through operator intervention is sufficient.

2.10.4 Protection of Keys

This section covers protection of CSPS. Basic defense against key tampering or theft is controlling the access to the module itself.

1. Controlled physical access to the module & Apollo chassis in general has been assumed in our model.

2. The operating environment on the module is controlled and it doesn't allow multi-user (concurrent or otherwise) access to the module. Please see the section titled 'Operational Environment' for further details.

3. In addition to the measures discussed above, special attention is paid to CSPS which are plain-text keys in our case. The FAK (that is used for AES256-CBC decryption of HEK, HDK coming from the System card) is kept in an obfuscated form in the RAM. This key is only 'un-scrambled' for the duration of API call to decrypt HEK or HDK.

4. The hardware AES keys (i.e., HDK, HEK) for encryption and decryption are not stored in firmware in the plaintext form. These are programmed into the FPGA. Moreover these keys are ephemeral and rotated frequently, which greatly diminishes their value from attackers' perspective. By the time a key can be retrieved through a 'brute-force' technique, the system would expire and discard that key through the key rotation process.

2.10.5 Storage of Keys

The following table covers each key, its type and location of storage in TR10_4EN.

Sr. No.	Key Type	Storage Format	Storage Location & Description
1	AES256-CBC key for 'backplane' decryption	Key is obfuscated and stored as scattered fragments in memory.	IO application firmware image.
2	AES256-GCM keys for datapath encryption or decryption	Encrypted using AES256-CBC by the System card.	Stored in the encrypted form as a part of a configuration file on the 'RAM-disk' of TR10_4EN.
3	AES256-GCM keys for datapath encryption or decryption	Plain-text	Loaded in 'Sitar' FPGA.

Table 7: Key Storage

2.11 Self-Tests

The following section covers various self-tests. An important thing to note is that the module doesn't perform its normal operation (transponder) before or during the self-test phase. No data is output from the data-output interfaces.

Moreover normal operation only resumes once all the tests pass in a given self-test run. If a self-test fails, the System card powers down the TR10_4EN module.

2.11.1 Power-up Self Tests

When the module is powered up a set of firmware and hardware tests are automatically launched. If any of the tests fail, appropriate management alarms are raised and 'green shutdown' of the card is performed. Therefore the card may not perform its normal operation (i.e., provide encryption services) if it has failed a self-test. The operator must either reset or power-cycle the card that has failed any self-test. If the failures are persistent, the card may have to be replaced.

A card can only resume normal operation if it passes all tests. If all the tests pass, the status of the module in the management plane on the System card is reported as 'Up' without any self-test alarms.

2.11.1.1 Integrity Check

1. Upon Power-Up, the boot ROM application on TR10_4EN downloads the OS image (vxWorks), a Sitar FPGA image & a firmware application image from the System card. These are stored as files on the RAM disk.

2. The firmware application starts and first computes CRC32 checksum on both the OS, FPGA & application images from the RAM disk. The application also has access to the 'reference' CRC values of these images. These reference values are stored in a separate file on the System card. This file is fetched in the previous step along with OS, FPGA and application images.

3. If there is a mismatch between the computed and reference values of an image file, the integrity check fails. At that point the module reports the failure to the System card.

4. The System card performs a green shutdown of the module. Power to the module is cut until there is an operator intervention to restart the card or the card is replaced in the slot.

2.11.1.2 Self-Test on Firmware AES

As explained earlier there is a firmware implementation of AES256-CBC on the module that decrypts CSPs received from the System card. This implementation is tested for its correctness using known-answer-test (KAT) approach.

It should be noted that before running the AES self-test the module runs the integrity check previously discussed in order to gain confidence in the integrity of the firmware image. As a part of the AES self-test, the firmware decrypts a pattern for which the answer is known. The result of the decryption is compared with the known answer. A mismatch between the result and known answer implies failure of this self-test.

Any AES firmware test failure leads to a notification for the System card. The System card then raises an appropriate alarm and performs green-shutdown of the card. In the green-shutdown power to the module is removed. The module remains in the power down state until there is an operator intervention from the System card.

2.11.1.3 Self-Test on Hardware AES implementation

Another self-test performed by the module after power up is that of the AES hardware implementation in its 'Sitar' FPGA. This implementation is used to encrypt/ decrypt traffic through line ports of the module.

Both the encryption and decryption functionalities are tested separately and failure in either one of these leads to alarms and 'green-shutdown' of the card as described earlier.

The details of these self-tests are available in the 'SDD' (Firmware Design Document) accompanying our FIPS documentation set. In a nutshell a Known-Answer-Test (KAT) approach is followed to test the hardware.

A standard pattern is provided as input to the encryption block in the FPGA. The encryption block performs encryption on this pattern. The results of the encryption are then compared with the answer known to the application firmware. The test is considered as successful if there is a match.

Similarly the decryption block is tested by decrypting a standard pattern whose answer is known to the application.

Any mismatch between the encryption or decryption results with their reference answers leads to reporting of failure to the System card which then raises an alarm and also performs green shutdown of the card.

2.11.2 Conditional Self Tests

None of the conditional tests mentioned in [section 4.9.2 of FIPS140-2] are applicable to TR10_4EN. When an operator resets a module the same set of self-tests that run upon power-on are triggered. It forces the module go through the entire sequence of image-integrity-check, firmware-AES and hardware-AES tests described previously.

2.11.3 Summary of Self Tests

Sr. No.	Test Name	Error Indicator upon Test Failure
1	Image Integrity	Specific alarm (***imgIntegrityFail***) on the System card, power-down of TR10_4EN module that has failed the test.
2	AES-CBC Decrypt KAT (firmware AES)	Specific alarm (***swCryptoSelfTestFail***) on the System card, power-down of TR10_4EN module that has failed the test.
3	AES-GCM Encrypt KAT (hardware AES)	Specific alarm (***hwEncSelfTestFail***) on the System card, power-down of TR10_4EN module that has failed the test.
4	AES-GCM Decrypt KAT (hardware AES)	Specific alarm (***hwDecSelfTestFail***) on the System card, power-down of TR10_4EN module that has failed the test.

Table 8: Self-Test Summary

If all the tests pass the status of the module in the management plane on the System card is reported as 'Up' without any self-test alarms.

2.12 Design Assurance

Best design practices are followed & various measures are taken to assure sound firmware and hardware design of TR10_4EN:

1. We use object oriented firmware design practices and use contemporary tools such as UML (Unified Modeling Language).

2. We use a well-known tool called Perforce (P4) for software configuration management (SCM) to assure reproducibility of our firmware build. Similarly we use a well-known open source tool called reviewboard for peer-code review process.

3. For programmable logic we use Synchronicity for SCM.

4. We use a well-known document management system (DMS) called Sharepoint to document, review and modify our design.

5. Each key component of our firmware implementation is peer-reviewed, unit-tested and it later goes through multiple rounds of feature, stability and regression testing for quality assurance purposes.

6. Similarly the hardware (programmable logic as well as physical board) goes through several rounds of qualification before the release. The hardware meets US-FCC and other relevant standards from certification and statutory bodies in countries where it is marketed.

7. The cryptographic algorithms have passed CAVP as a part of FIPS140 certification.

2.13 Mitigation of Attacks

This section is not applicable. The module does not claim to mitigate any additional attacks in an approved FIPS mode of operation.

3 Secure Operation

TR10_4EN always only operates in the FIPS approved mode. It doesn't require any explicit configuration/ trigger to run in the FIPS approved mode. Non FIPS approved modes are not supported.

3.1 Module Installation

Installation of the module is fairly straightforward. All the operational code and CSPS come from the System card. The only 'state' stored on the card is in its PROM. PROM contains information such as board serial numbers, hardware revision etc. and the Boot Flash (aka Boot-ROM). The Boot Flash allows the module to load other operational images from the System card.

A TR10_4EN module can be provisioned from the System card either through EMS or through CLI. A CLI based command sequence will be as follows:

1. Configure a slot to hold TR10_4EN module using '***set chassis slot <slot> <card>***' command.

2. Insert a TR10_4EN module in the slot OR if you want to run the initialization test again on an inserted module issue '***run request chassis power reset***' command.

3. The module will initialize and it will transition through various states described in the FSM document.

4. From the CLI, the module status will transition from '*down->initializing->configuring->up*'. You can see this change by repeatedly running the '***run show chassis status***' command.

5. Details related to various hardware components of the chassis can be seen by running '***run show chassis hardware***'. Similarly various alarms can be monitored through '***run show chassis alarms***'

command. The '***run show chassis configuration***' is another useful command to see details related to various ports of the chassis.

When the module is 'up' it's ready to run services in FIPS approved mode. The services need to be explicitly configured for a given pair or ports. All the configuration parameters related to secure service are handled by the System card. TR10_4EN gets service related parameters (viz. HEK, HDK) through control messages sent from System card.

3.2 Initial Key Loading

No CSP/ key is loaded statically on a TR10_4EN module as a part of manufacturing process.

The firmware-AES key (FAK) is embedded into the firmware operational image in an obfuscated form. This key is retrieved by the firmware only for the purposes of decrypting the HDKs and HEKs.

The traffic encryption/ decryption keys (HDKs & HEKs) are established not by TR10_4EN but by the System card. These are then sent to TR10_4EN using AES256-CBC encryption. TR10_4EN firmware decrypts and then loads these keys in the Sitar FPGA.

It should be noted that HDK or HEK are related to service provisioning and will not be seen on an idle card. FAK on the other hand is present (obfuscated) as long as the firmware is running on the card.

3.3 Administrator Guidance

This section covers instructions related to configuration of the module, security events and any assumptions related to security of the module.

1. Security of the System card and that of backplane Ethernet network is assumed in the overall security model.

2. The System card maintains persistent configuration for each TR10_4EN module in the chassis. It also maintains various service related parameters.

3. Administrators need to look for service-affecting alarms reported by the System card. These alarms include various failure events (including the self-test related) on TR10_4EN. Any kind of failure in self-test leads to a service affecting alarm on the System card which then performs 'green-shutdown' of the module.

4. Administrators can initiate 'chassis power reset' operation to restart a module that has entered green shutdown state. This procedure is described in product documentation that's shared with end users.

5. In case the module's power is lost and then restored, a new key for use with the AES GCM encryption and decryption shall be established.

3.4 Security Officer

1. TR10_4EN only resumes a configured cryptographic service when it gets the necessary key(s) (HEK or HDK) from the System card.

2. The System card needs control plane/ IP connectivity with remote peers in order to establish HEK and HDK. IP address of the remote peer corresponding to each provisioned line port of a given TR10_4EN module must be configured correctly.

3. Failure to establish control plane connectivity with remote IP peer reflects in the operational details provided by the System card.

3.5 Documentation Note

As per the requirements of the Vendor Evidence Document, we shall explain the relationship between various documents related to TR10_4EN including this one (the Security Policy).

1. Basic documentation starts with MRD (Marketing Research Document), which is an ECI internal document containing marketing data, competitive analysis and product vision.

2. The SRQ (System Requirement Document) takes MRD from concept to product realization level. It defines the physical module, its ports and services including the firmware features.

3. The System Design Document (SDD) is a top level architectural document for TR10_4EN product. It covers different pieces of firmware and hardware. This is a proprietary document but it's been shared with the certification agency under an NDA.

4. A set of detailed design documents for different firmware and hardware components of the product are then developed by domain experts. These documents include HLD (High Level Design), EDD (External Design Document) and IDD (Internal Design Documents). These are proprietary, ECI-internal documents. These documents are written, reviewed and maintain as a part of our design assurance process.

5. The FSM (Finite State Model) and SP (Security Policy) address specific areas of the product for FIPS certification process. These are shared with the certification agency. The FSM & SP cover 'non-proprietary' information.

6. All of the above documents are controlled. Their revisions are tracked through Sharepoint DMS within ECI. Each document has a unique associated ID.

7. Vendor Evidence is a supplementary document for FIPS purposes. Its primary audience is the certification agency that can get specific reference/ pointers from other documents listed above.

3.6 Traceability & Identification

In a production environment different physical components and firmware versions must be identifiable in order to correctly deploy a service that provides security as intended.

3.6.1 Hardware Identification

All the FRU (Field Replaceable Unit) components contain ECI labels, hardware revision number and serial numbers visible on the parts. TR10_4EN contains a PROM (Programmable Read-Only Memory) that

stores this information which can also be retrieved remotely through the System card. This enables tracking of hardware parts. There is a firmware framework on system-card that exports this information to management applications remotely managing the card. [For example a CLI command '***show chassis configuration detail***' prints all the hardware details to an operator terminal].

TR10_4EN is comprised of 3 PCBs: Base-board & 2 mezzanine cards mounted on the top of the base board. One mezzanine-card is a power module and another mezzanine card has is CPU block (CPU, RAM, ROM etc.). Additionally the pluggable transceivers (XFP or SFP) on each port contain PROM data for their identification.

3.6.2 Firmware Traceability

There are 2 firmware bundles installed in an Apollo chassis. The first bundle (called 'MOS') provides Operating system related binaries for the System card. The second bundle (called 'Shadetree Bundle') contains the following:

1. Binaries of various firmware applications running on each card in the Apollo chassis. This is a superset of all binaries. A given chassis may only use a subset of these depending upon the types of card actually plugged into the chassis.
2. FPGA images required for various cards in the chassis (this includes images for TR10_4EN).
3. Operating system images (such as 'VxWorks.bin') required by various line cards in the chassis.
4. Checksum files for files (OS, firmware and FPGA) used by TR10_4EN card. These files are read-only. These are generated by ECI's release/ build machines.

Both the MOS and Shadetree bundles in the chassis are traceable. Firmware infrastructure on the System card exports the version information (build number, date, firmware branch etc.) to local or remote management applications. [For example, a CLI command '***show version detail***' prints all the relevant details.]

Acronyms

Keyword	Explanation
CAVP	Cryptographic Algorithm Validation Program
CLI	Command Line Interface
CMVP	Cryptographic Module Validation Program
CSE	Communications Security Establishment (Canada)
DH	Diffie Hellman (Key establishment Algorithm)
EMS	Element Management System
FAK	Firmware AES Key
FCC	Federal Communications Commission
FIPS	Federal Information Processing Standards
FPGA	Field Programmable Gate Array
FSM	Finite State Model
GPIO	General Purpose Input Output
HDK	Hardware Decryption Key
HEK	Hardware Encryption Key
IP	Internet Protocol
IS	In Service
NIST	National Institutes of Standards and Technology
ODU	Optical Data Unit
OTN	Optical Transport Network
RAM	Random Access (read, write) memory
ROM	Read Only Memory
SCM	Firmware Configuration Management(hardware AES)
SW-U	Software In Use
TAC	Technical Assistance
TR10_4EN	Coded name of the encryption module. **TR** stands for Transponder, **10** stands for 10Gbps line speed and **4EN** stands for 4 ports (2 client+ 2 line) encryption card.
UML	Unified Modeling Language