FIPS140-2 Non Proprietary Security Policy

ECI TM200EN Encryption Module

Hardware Part No: <TM200ENB, TM100_2ENB, TR10_12ENB>

Firmware Revision: <R9.0>

System Name	Apollo
Sub System Name	IO
Module Name	TM200ENB
Revision	1.6
Release Date	20-Oct-2021
Document ID	2EVYDPYSQPWJ-712128828-1359

Canadian Center For Cyber Security

http://www.ecitele.com

Contents

List of Figures

List of Tables

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
7	Recommendation for Pair-Wise Key-Establishment Schemes Using Discrete Logarithm Cryptography	NIST, US Dept. of commerce		SP800-56Ar3

1 Introduction

This document covers the non-proprietary security policy for ECI TM200EN Encryption Module. This document is prepared in the context of FIPS140-2 standard compliance of the module. ECI TM200EN Encryption Module operates as a line card in ECI's Apollo product line. In the networking parlance it's a transponder with multiplexing and encryption capabilities.

In the interest of brevity, we shall refer to this product as "ECI TM200EN" or simply "TM200EN" in the rest of this document.

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.

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 (CCCS) federal governments. ECI TM200EN Encryption Module 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 (**R9.0**) that meets **FIPS140-2 Level-2** 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	2
Cryptographic Module Ports & Interfaces	2
Roles, Services & Authentication	2
Authentication	2
Finite State Model	2
Physical Security	2
Operational Environment	N/A
Cryptographic Key Management	2
Self-Tests	2
Design Assurance	2
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 ECI TM200EN Encryption Module (hereby referred as 'the module'/ 'IO' 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 & acronyms tables.

2 Module Overview

2.1 System Level Block Diagram

The following block diagram provides a system level overview of transponder with encryption application. It covers the control plane as well as data-plane aspects of the module.

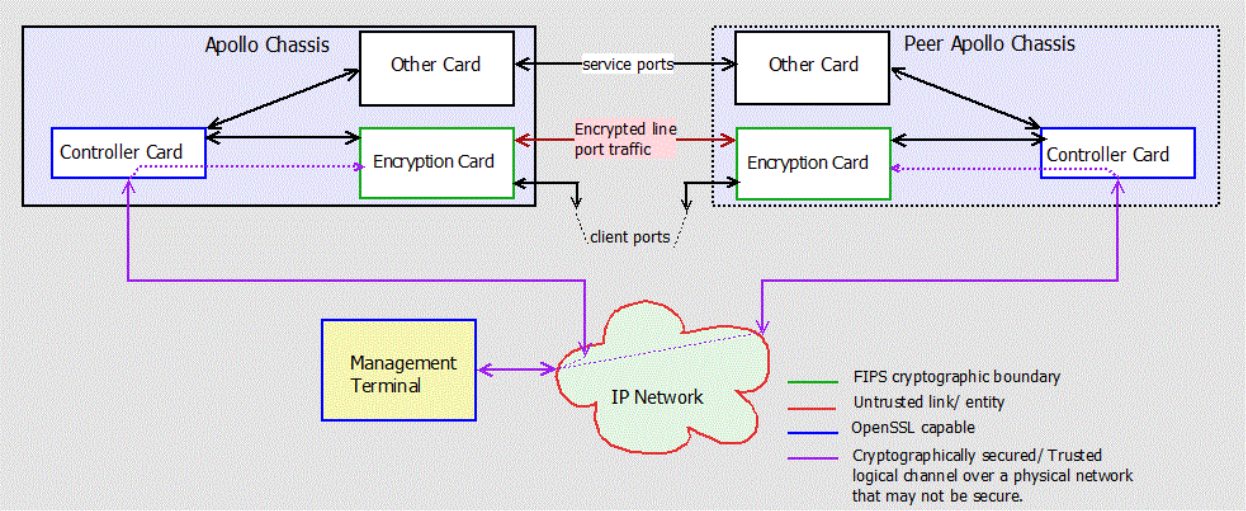

Figure 1: System Level Overview

The module basically acts as a transponder. It exchanges traffic between its client side ports and line side ports. Client side traffic is not encrypted whereas traffic on the line is usually encrypted. One or more client side traffic streams are multiplexed into a line side stream. Each client side stream is independently encrypted.

As shown in the diagram above, the module exchanges traffic in an encrypted form with remote peers reachable through its line-side ports. These remote peers are typically located at great distances and/ or reachable across third party networks, which warrants traffic encryption. 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. The module establishes ephemeral encryption keys with remote peer-module through signaling. There is TCP/IP connectivity between peer modules for establishing keys.

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

ECI TM200EN Encryption Module is a '2U' sized card that fits into ECI's 96xx Apollo chassis and occupies two adjacent slots. ECI is seeking FIPS certification only for this module in its upcoming release (R9.0). Other card types or the Apollo chassis are not part of this FIPS certification. The module is shown in the following pictures:

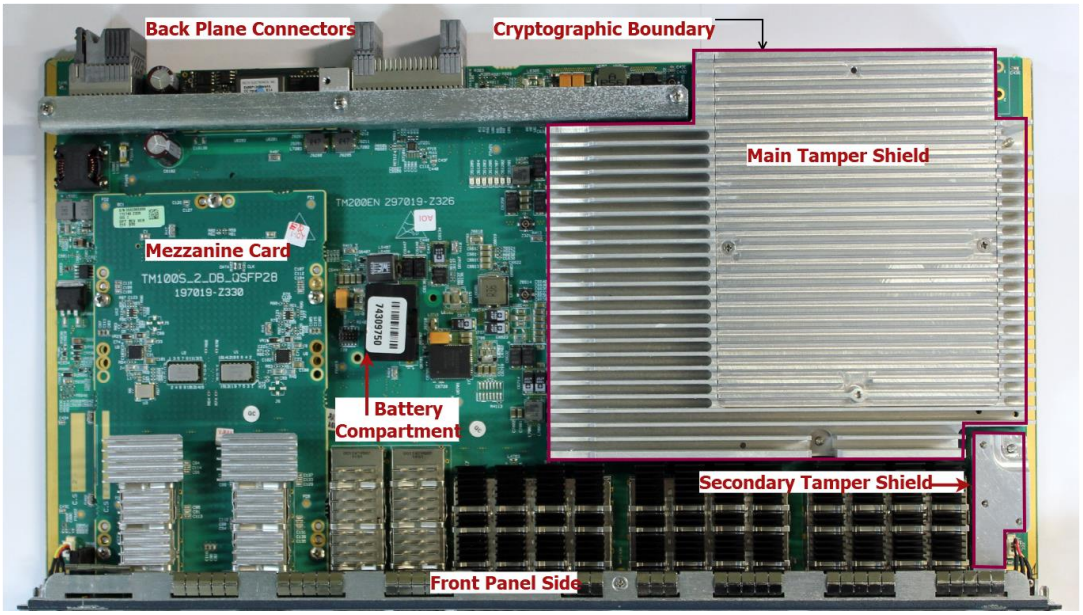

Figure 2: ECI TM200EN Physical Module (Top View)

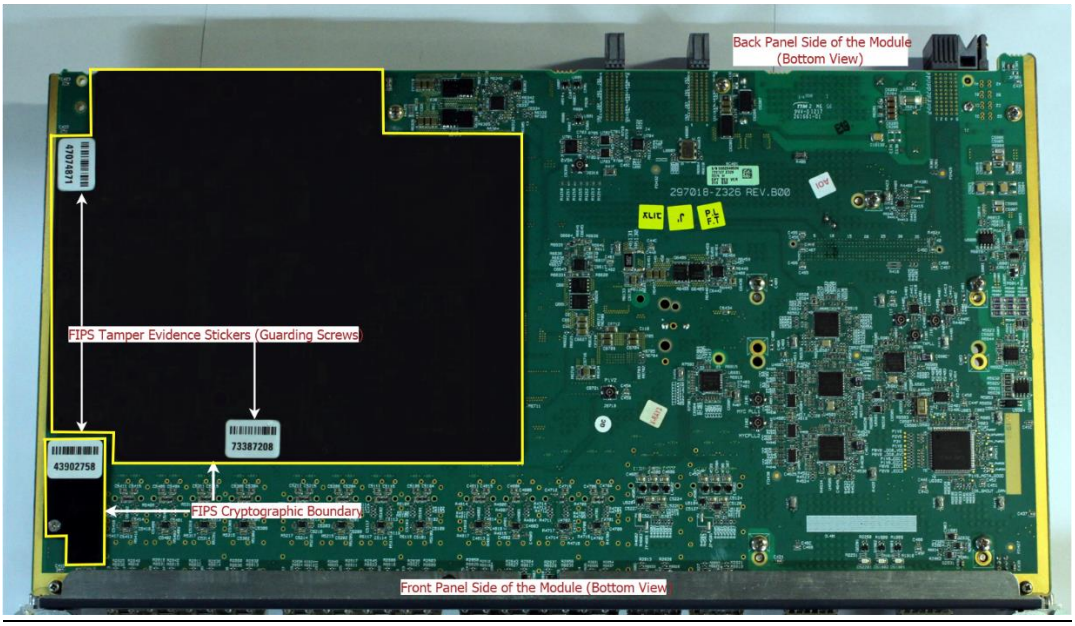

Figure 3: ECI TM200EN Encryption Module (Bottom View)

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 the Apollo backplane and also receives commands/ control information & parameters through its backplane Ethernet interface.

Physically the module comprises of a base board fitted with a FIPS level-2 compliant main tamper shield and a small secondary tamper shield serviceable only at ECI facilities.

The following critical components are protected underneath this shield:

1. CPU (Freescale HCPM T1042 with dual PowerPC e5000 cores) & RAM. The ROM is embedded into T1042 and provides SecureBoot functionality.
2. Hardware Encryption Engine and Framer for line & client ports
3. Tamper monitoring circuitry
4. A flash memory device
5. Programmable logic devices providing glue-logic, small embedded memories.

The secondary tamper shield guards some PCB signals useful for running board diagnostics & maintenance but which are also deemed as potentially sensitive.

2.2.1 Non Security Related Components

All components outside the tamper shields are deemed non-critical from a FIPS perspective. These include pluggable optics, back-plane connectors, a serial port, face-plate, power-supply, programmable glue-logic, LEDs etc. Malfunctioning of one or more of these components will likely affect the service availability but will not compromise the cryptographic integrity (such as protected CSPs or algorithms) of the module.

The following depiction covers the main board layout including the tamper cover.

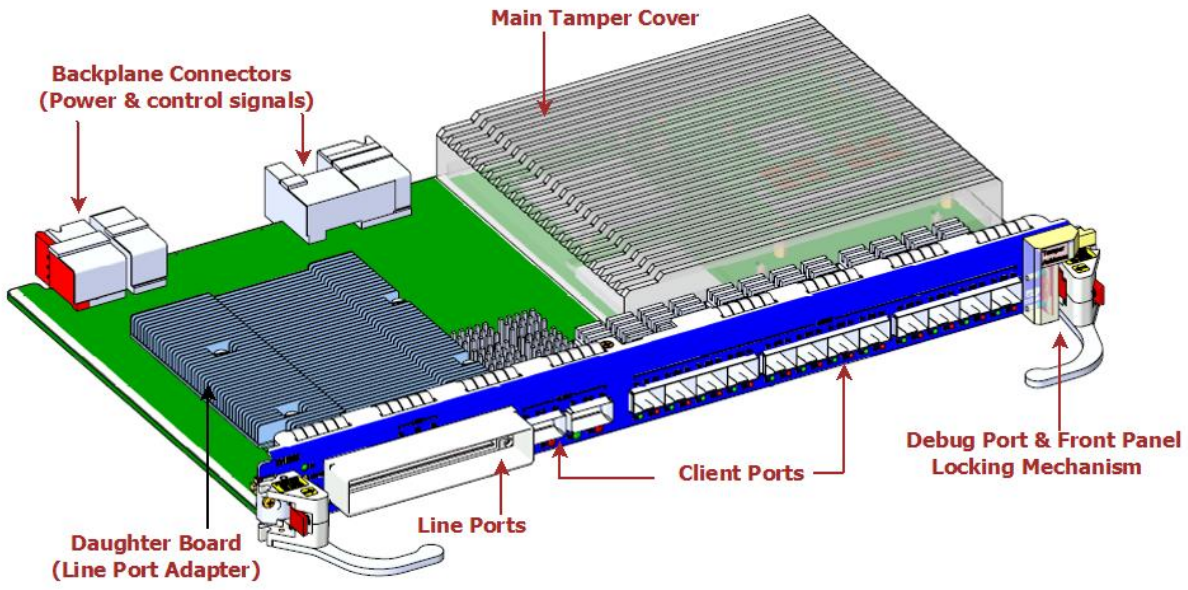

Figure 4: ECI TM200EN Board Assembly

2.3 Module Ports & Interfaces

2.3.1 Data Ports

The module has two types of data ports: client-ports & line-ports. There is a provision for 20 client ports and 1-2 line ports. ECI TM200EN Encryption Module comes in three variants TM200ENB, TM100_2ENB & TR10_12ENB.

1. TM200ENB: There is a single 200Gbps line port
2. TM100_2ENB: There are 2x100Gbps line ports.
3. TR10_12ENB: 6x10Gbps client ports connected to 6x10Gbps line side ports.

In all of the above cases, the encryption engine working at 10Gbps streams is the same. It's not mandatory to populate all the client or line ports. On the contrary, the pluggable optics allows customers to only populate a subset of available ports. Additionally, this scheme allows a mix and match of different types of signals (OTN/ Ethernet/ SDH/ Fiber-Channel etc.) in a given deployment. The firmware image as well as the main encryption/framer-engine, parts underneath the tamper shields are identical between the variants.

The following depiction covers front panel arrangement of various ports in ECI TM200EN variants.

Figure 5: Ports and Interfaces on TM100_2ENB

Figure 6: Ports and Interfaces on TM200ENB

2.3.2 Debug Port

The debug port is only required for the initial CO[1] password configuration when a card is received from factory. Beyond this initial step, the debug port may be sealed using a FIPS compliant seal as a security measure. That way any unauthorized access to the debug port can be monitored. However covering the debug port is not mandatory for FIPS compliance.

Each time when a card is powered up, a firmware menu is presented through the debug port. The CO can interrupt the boot process, provide the CO password and run privileged commands. After the boot phase a regular login prompt can be seen through the debug port. Regular (non CO) operators can perform login authentication using their password.

[1] CO: Cryptographic Officer as defined in the FIPS standard.

In other words, beyond the initial CO password configuration, authentication (by the CO or by operators) is always required in order to perform CO related tasks. The CO and operators have different roles and different access privileges to the module. In short, the CO is responsible for the physical security of the card and managing its persistent CSPs; while operators are responsible for service configuration, monitoring and log collection. Operators do not have the access rights to modify the CSPs.

2.3.3 Management Ethernet

ECI TM200EN Encryption Module is a line card in Apollo chassis that is configured & monitored through the System card (sometimes called the RCP: Route Control Processor in ECI literature) in the chassis. This system card sends various image files, boot parameters as well as command and configuration parameters to each ECI TM200EN Encryption Module in its chassis.

There is an intra-chassis, Ethernet link running between the System card and any line card module. This Ethernet link is not accessible to end users; it's part of the backplane signals of the Apollo chassis.

SSH Interface can be used to load the initial CSPS (instead of the debug port). Under normal operating conditions, SSH is not required since there is a trusted control messaging channel running between the embedded applications running on the system card and ECI TM200EN Encryption Module for service provisioning and monitoring.

2.3.4 Summary of FIPS140-2 Interfaces

Interface/ Port	Interface Type	Description
Client Port 1-20	Data Input & Output	These ports are located on the front panel. Users may provision all or only a subset of client ports at a given time. Each provisioned client port runs data traffic to a customer premises without encryption. Traffic from a client port is cross-connected to one of the line ports of the module.
Line Port 0-1	Data Input & Output	Depending upon the hardware flavor ECI TM200EN Encryption Module has 1 or 2 line ports. These ports are located on the front panel. Each line port exchanges OTN data traffic with a remote-peer. This traffic is encrypted (except for the bypass mode) using a FIPS approved algorithm.
Debug	Control Input	Used to configure CO password and CSPS.
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 yellow 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
Security LED	Status output interface	This is a multi-colored LED that reflects FIPS/ security related status of the card.
Logical Control Channel	Control input	This is a logical control messaging channel that uses the management Ethernet physical link between the system card and the module.
Backplane Control Signals	Control Input	The system card can perform certain control operations on the module 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 pluggable optical transceivers. When a data port is not provisioned, its corresponding transceivers may not be plugged-in.

2.3.5 Physical versus Logical IO mappings

The following table shows relationships between physical and logical entities that are input/output interfaces of the module.

Physical Entity	Interface Type	Description of the corresponding Logical Entity
Client Ports 1-20	Data Input & Output	A provisioned client port maps to a logical interface such as ODU or Ether-VLAN depending upon the encapsulation. This logical interface is cross-connected to another logical interface from a line port.
Line Ports 0-1	Data Input & Output	Each provisioned line port has an associated logical interface that is used for cross-connections. A cross-connection represents a service instance.
Debug	Control Input	It's required only for configuring the initial CO password. It can also be used to load CSPS (private key, certificates).
LED(s)	Status output	Each LED is controlled through one or more register bits located in programmable logic devices.
Ethernet line to system card	Control input & Status output	There is a physical Ethernet link connecting each module to the active System card in its chassis. There is a logical, TCP/IP based control channel corresponding to this physical link. S/W application running on the System card uses this logical channel to control the module. There can be up to 2 system cards in a chassis (active & standby). Only one is active at a given time. There is a point-to-point Ethernet link between each system card and a given module. In other words the module has 2 management Ethernet ports, only one of which can be active at any given time.
Backplane Control Signals	Control Input	An active system card can perform basic module control operations (e.g. reset, power-control, board-presence-detection) through these signals.

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 the 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 starts executing.
FAIL	card-level-status	This is a red colored LED, which is turned on when there is

LED Name	Scope (port/ card)	Description
		a service affecting, major failure on the card.
Security	card-level-status	This is a dual (green & red) colored LED. It indicates tamper/ card-security status through various color and flash-rate patterns. These patterns and the corresponding state of the module are made available to customers through user manuals. The salient patterns defined are: 1. Solid-green: Board is fully initialized, passed all self-tests and running in FIPS compliant mode. 2. Solid-red: There had been a tamper event with the board. The board must be returned to ECI as it fails to initialize.
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). It's controlled by hardware.

Table 4: LED Roles

2.4 Roles & Services

2.4.1 Roles

1. In production environments operators do not directly connect to the module and manually enter commands. Instead s/w application from the System card sends proprietary control messages and retrieves status information from the module across the backplane Ethernet interface (i.e. the 'control-channel').

2. The control channel can also be used by CO (upon authentication) to load cryptographic keys and certificates into the module. These keys and certificates are CSPs in the FIPS parlance.

3. Additionally, the s/w application running on the module uses the control-channel to establish shared-secret / encryption-keys with remote peer modules. It exchanges messages with remote peers to establish ephemeral encryption keys using FIPS approved algorithms.

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

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, Configure Service	User (unauthenticated)	None	Input: Control messages from system card to the module Output: replies to the above control messages. Interface: Ethernet control-channel
Perform Self-Test	User (authenticated)	Public, Private keypair, CO password, Key-Encrypting-Key (KEK) All the above are accessed in read-only (R) mode.	Input: System card initiated reset of the module in a given slot. Output: Hardware reset is applied to various module components. Then the module runs various self-tests. Interface: Dedicated control signal from system card.
LED status	User	None	Input: None Output: The LEDS mounted on the front panel of the module are output interfaces that provide visual status information about the state of the module and configured services.
AES key negotiation	Crypto Officer	Public, Private keypair, CO password, Key-Encrypting-Key (KEK), Self & CA certificates All the above are accessed in read-only (R) mode.	Input: Service configuration Output: Software application running on the module exchanges messages with remote module to establish ephemeral encryption keys for the data path to use. During this messaging, FIPS compliant algorithms are used for peer authentication (e.g. RSA) and shared secret establishment (e.g. DH). Interface: Ethernet control channel
Datapath Encryption	User (h/w component)	HEK key in RX	Input: Client side traffic Output: Client-side payload exiting each line port is encrypted using the AES256-GMAC algorithm. Interfaces: Client and Line ports

Service	Role	CSP & Type of Access (R: Read, W: Write, X: Execute)	Details
Datapath Decryption	User (h/w component)	HDK key in RX	Input: Line side traffic Output: Payloads destined for client ports that are received on a line port are decrypted using the AES256-GMAC algorithm. Decrypted payloads are then forwarded to appropriate client ports. Interfaces: Client and Line ports
CO Password Configuration	Crypto Officer	CO password is accessed in RW mode.	Input: CO uses the debug port of the module to load/ modify CO password. Output: Module stores the password in its protected, persistent memory. Interface: Debug Port
User Initiated Zeroization	User (unauthenticated)	HEK, HDK are accessed in RW mode.	Input: User Initiates module reset or removes previously provisioned service. Output: The module removes all or specific encryption keys depending upon the user initiated operation. Interface: Ethernet control channel
Selective Bypass	User	None	The module can selectively disable encryption on a subset of service instances (i.e. cross-connects), while the encryption service may be active on other service instances.

Table 5: Summary of Services Provided by ECI TM200EN Encryption Module

A subset of the above scenarios involves authenticated-user or the CO (Crypto-Officer). In such scenarios, the concerned user or CO need to authenticate themselves before gaining access to the card. Authentication mechanisms are covered in the following subsection.

In each of the above service scenarios, CSPS (such as the private-key, CO-password, or traffic-encryption keys) are not revealed through any interface under any circumstances.

2.5 Authentication Mechanism

The module complies with role-based authentication requirements for FIPS level-2 module. The card supports following roles and modes of authentication:

1. **CO (Cryptographic Officer)**: CO Is the only person authorized to load persistent CSPs into the module.
    a. In order to load/ modify these CSPs, the CO must provide a valid CO password.
    b. The CO password is originally assigned during card installation. Later on the CO password can be modified provided the CO can provide the current password.

c. CO can load CO password, the {Public, Private} key pair & X.509 certificates into the module.
d. Only a single CO can be active at any given time (i.e. no concurrent operation). This simplifies the implementation and security handling. A new CO shell is launched and CO credentials are entered through it, the previous active CO shell session is terminated.
e. The CO password must be [8, 20] characters in length, with at least 1 special character, 1 upper-case and 1-lower case. With 92[2] possible values per character, it provides adequately low odds of a random match as per the FIPS specification.

2. **System-card software (User/ operator)**: The system card software is responsible for monitoring module health, FIPS status, service status and provisioning services.
   a. It uses proprietary messaging format to exchange information with the module.
   b. These access are made over the trusted intra-chassis Ethernet control-channel.
   c. System-card software may not modify/ substitute/ disclose CSPS on the card.
   d. Only one system card/ application instance can manage an ECI TM200EN Encryption Module at any given time. Concurrent access is not required and not supported.
3. **Human operator (User)**: Human operators can log in to the Linux OS running on the module either over a SSH (Secure Shell) connection or through the console port (i.e. the debug port). They must authenticate by entering a valid operator login and password.
   a. Operator access is supported to collect debug information under abnormal conditions. Operators may not modify CSPS on the card. Multiple human operators can concurrently log into the card and monitor the cad state/ collect debug data.
   b. The human operators however do not have access to CSPS or other operations that are privileged and only allowed for the CO.
   c. The operator password is at least 8 characters in length, and allows 92 different values per characters. This provides sufficiently low odds of random password match.

Each initialized FIPS module owns {public, private} key pair. Only the CO can modify it. The public key from this key pair can be read later by CO or Users. However, the private key is never revealed to anybody through any interface. Similarly, the ephemeral encryption keys for traffic encryption/ decryption are never revealed to the CO or any other user.

Authentication information such as operator login is not accessible unless the operator can perform successful login. The module never reveals the CO password or private-key to any users, including the CO. CO can only gain access to a module if the password they enter matches with previously configured value.

2.5.1 Persistence of Authentication

Any of the authentication modes described above are not persistent across module reboot/ reset or power-cycle operations. It must be performed again after any such operation as the results from previous authentication are erased after reboot/ reset or power-cycle.

2.6 Finite State Model

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

[2] 26 upper-case, 26 lower-case and 40 additional characters provided by 20 keys on ASCII keyboard (with or without SHIFT) leads to 26+26+40 = 92.

2.7 Physical Security

ECI TM200EN Encryption Module functions as a line card in ECI's Apollo chassis. It falls under the category of '*multi-chip embedded cryptographic modules*'. It's FIPS140-2 level-2 compliant. In that context:

1. The module provides evidence of tamper through multiple means:
    a. Primary tamper-evidence mechanism is through seals factory-applied across critical surfaces of the module. A tamper event is expected to irreversibly alter their visual and/ or mechanical characteristics.
    b. There is a FIPS compliant tamper cover that is mated (and sealed using the above material and other means) with the board. This cover prevents visual/ electrical or mechanical access to sensitive components on board.
    c. The module has embedded electronic tamper detection and response circuitry.
        i. This circuitry activates instantaneous tamper response if there is tamper event while the module is powered up.
        ii. Upon tamper normal operation of the h/w and firmware is immediately halted. Additionally, access to the private key stored on board is permanently removed.
        iii. An additional, persistent hardware marker is set which prevents initialization or re-installation of the module.
        iv. The module under these conditions must be returned to factory. Power-cycling the module doesn't restore it to working condition.
        v. The above features exceed FIPS level-2 security requirements that the module is certified for. Nevertheless these are there as additional security features.
    d. If there is an attempted tampering while a module powered down, there is battery backed electronic circuitry to remember this event. If the module is powered up subsequently, it fails to boot without CO's authentication & consent.
2. The module also has a locking & sealing mechanism that prevents removal from its slot after installation. This is achieved as follows:
    a. After seating the module in its chassis, a mechanical cover is screwed over the front panel.
    b. Then a FIPS compliant tamper-detection seal is applied across the cover. This prevents unauthorized removal of the module from its slot without tamper detection.
    c. The above mechanism is an additional security feature of the module and is not a requirement for FIPS level-2, and therefore hasn't been covered by the FIPS certification.
3. Tamper seals are encoded with encoded characters and their inventory is carefully tracked by the CO. Additional tamper seals can be ordered (part# X93617) separately from ECI if necessary.

4. To summarize the module meets or exceeds FIPS level-2 tamper detection requirements.

The following pictures cover physical-security & tamper-evidence features of the module.

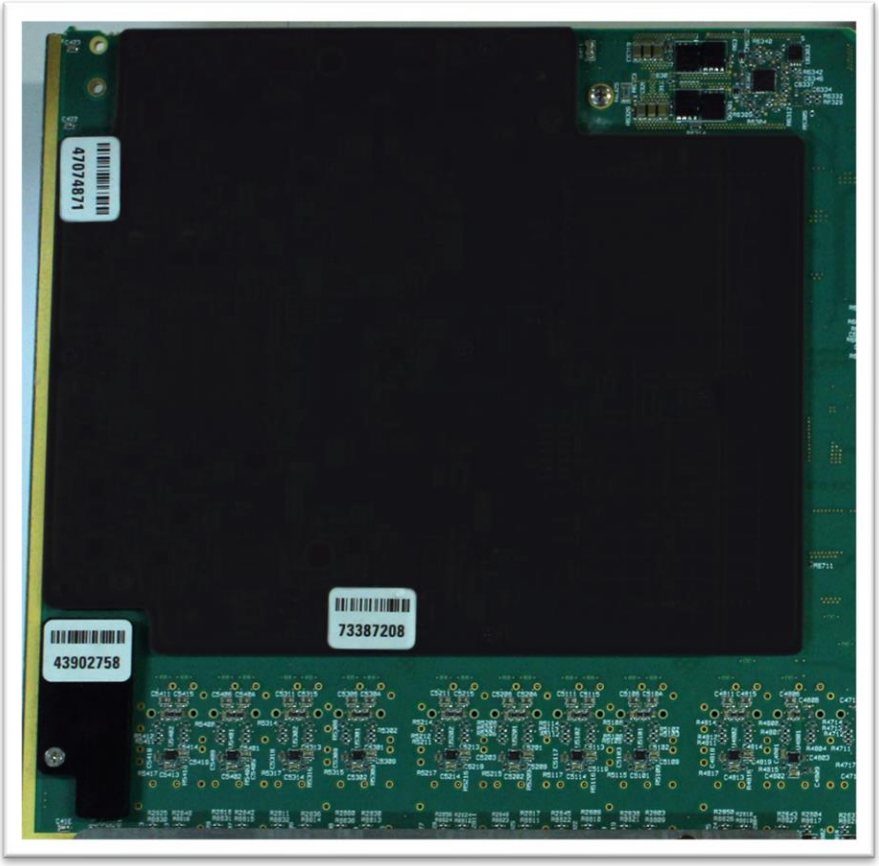

Figure 7: Tamper Evidence Seals Applied Across Critical Areas of the PCB (rear)

Figure 8: Metallic Tamper Shield Sealed and Mated with the PCB (front)

Figure 9: Front Panel Before Installing Pick Resistant Lock

Figure 10: Front Panel After Installing Pick Resistant Lock & Tamper Seal

2.8 Operational Environment

The module runs ECI customized version of embedded Linux operating system (branded as MOS: Magnum OS). MOS doesn't support the ability to add/ delete/ modify applications dynamically (i.e. in production environments). ECI releases and controls the operational firmware image.

The module (i.e. Boot-ROM application) fetches MOS image file from the System card, verifies its digital signature and then executes it. MOS is a limited operational environment from FIPS perspective. Users may not add/ remove/ modify it. Hence it's considered as 'limited operational environment' from a FIPS' perspective. Therefore the clauses in section 4.6.1 of FIPS140-2 do not apply.

In production environment, the System card uses a proprietary messaging mechanism to interact with user space application processes running on the module. This mechanism facilitates configuration and monitoring of encryption services on the module, but doesn't alter the MOS itself.

2.9 Cryptographic Key Management

This section covers various cryptographic keys stored/ managed by the module. All the keys on board are stored in different kinds of memory devices that are shielded by the main tamper cover. Tampering with these parameters without generating tamper-indications is not feasible.

1. There is an AES256 key for each provisioned client port running in encryption mode. There are up to 20 client ports per module. We shall call it **HEK** (Hardware Encryption Key). This key is used to encrypt traffic stream arriving from client port before it's sent out of a line port. The line port may multiplex multiple such streams from different client ports.

2. Similar to the HEK, there is **HDK** (Hardware Decryption Key). It's used to decrypt the traffic stream from line port before its transmitted out of a client port.

3. Both HDK and HEK are negotiated with a remote peer module through a proprietary messaging mechanism. The negotiation is based on FIPS approved algorithms for authentication (e.g. RSA & SHA256) of the peer and secure key-establishment (e.g. Diffie-Hellman). HEK and HDK are ephemeral and non-persistent. These are discarded periodically when fresh keys are negotiated with the peer. The keys are also discarded when the corresponding service is deleted or when the module gets a reset/ restart.

4. There is a unique {**M_private, M_public**} RSA key pair per module.

    a. The RSA key pair is encrypted using the AES256 cipher and stored in a persistent memory, under the tamper shield. It's stored in an encrypted form, encrypted using an AES256 key called **KEK** (Key Encrypting Key).

    b. The **KEK** is generated using onboard random number logic and is stored in an obfuscated form in another persistent memory under the tamper shield. KEK is destroyed automatically if the onboard electronics senses a tamper event. Tamper sensing circuitry monitors the module regardless of whether it's receiving mains power supply.

    c. Corresponding to M_public, there is **M_cert** (an X.509 certificate) stored in a flash memory onboard. M_cert is the certificate issued to a specific module by some CA (Certification Authority). This certificate is used for peer authentication. The certificate contains M_public along with a digital signature generated by the CA (Certification Authority).

    d. There is also a **CA_cert**, certificate of the Root-CA stored in flash that's used to authenticate certificates sent by various remote peer modules.

    e. As of R9.0 the module supports RSA-2048 bit modulus.

5. There is a CO password: **CO_pwd**, that's stored in another persistent & shielded memory on board. The CO must input a password, matching with the stored password before any privileged operation (including CO_pwd modification) is performed. The stored CO_pwd may not be retrieved or printed through any means. The module must be returned to factory if CO_pwd is lost.

6. There is a secure boot-ROM with an embedded RSA public key: **Boot_public**. This key is used for verifying the digital signature associated with software operational image that the module fetches from the System card as a part of the boot process.

The following table summarizes various keys and their roles.

Parameter	Type	Accessed By & Associated Service	Lifecycle, Input/ Output (I/O)
HEK	AES256 key	Module application, Datapath Encryption	Ephemeral, rotated periodically by the s/w application I/O> N/A. Dynamically generated in the module itself using cryptographic algorithms (DH).
HDK	AES256 key	Module application, Datapath Encryption	Ephemeral, rotated periodically by the s/w application I/O> N/A. Dynamically generated in the module itself using cryptographic algorithms (DH).
M_private	RSA private key	Module application (read), CO (write) Self-Test (R), AES-Key negotiation	CO can install & modify as a key-pair. I/O> Input by the CO during installation or auto-generated by the module using RNG. Never output.
M_Public	RSA public key	Module application (read), CO (read, write) Self-Test (R), AES-Key negotiation	CO can install & modify as a key-pair. I/O> Input by the CO during installation or auto-generated by the module using RNG. Exchanged through the corresponding X.509 certificate for authentication.
KEK	AES256 key	Module application Self-Test (R), AES-Key negotiation	KEK is used to encrypt M_private. I/O> N/A. Dynamically generated onboard and stored under tamper cover in obfuscated form.
M_cert	X.509 certificate	Module application (read), CO (read, write) AES-Key negotiation	CO can install & modify I/O> Input by CO during installation, exchanged with peers across the network for peer to peer authentication.
CA_cert	X.509 certificate	Module application (read), CO (read, write) AES-Key negotiation	CO can install & modify I/O> Input by CO during installation, used to authenticate peer certificates. Not required to be output.
CO_pwd	[8..19] octet text	CO (write), Hardware (write, read-match) CO password configuration	CO can create & modify I/O> Input by CO during installation, used for CO authentication. Never output.
Boot_public	RSA public key	Boot-ROM (read), CO (write) Firmware authentication	Factory installed, Future upgradable through trusted means. I/O> Input at the factory. Used for signature verification during the boot process. Not required to be output. Protected under tamper cover.

Parameter	Type	Accessed By & Associated Service	Lifecycle, Input/ Output (I/O)
DRBG Entropy Input String[3]	8192-bit value	Embedded RAM	Plaintext I/O> N/A. Dynamically generated and destroyed within the module itself.
DRBG Key Value	Internal DRBG state value	Embedded RAM	Plaintext I/O> N/A. Dynamically generated and destroyed within the module itself.
DRBG 'V' Value	Internal DRBG state value	Embedded RAM	Plaintext I/O> N/A. Dynamically generated and destroyed within the module itself.
DRBG Seed	384-bit value	Embedded RAM	Plaintext I/O> N/A. Dynamically generated and destroyed within the module itself.

Table 6: Summary of Module's CSPs

2.9.1 Approved Cryptographic Algorithms

Following is the list of algorithms used by the ECI TM200EN Encryption Module application.

Algorithm	Standard	Modes & Key Sizes	Function Performed	Validation Number
HMAC-SHA256	FIPS 198-1	N/A	Key derivation function	HMAC-3765/ CAVP-2574
SHA256	FIPS 180-4	N/A	Key derivation function	SHS-4532/ CAVP-21371
RSA	FIPS 186-2 FIPS 186-4	2048 bits modulus	Digital signature verification, challenge authentication.	RSA-3041/ CAVP-26499
ECDH Component CVL	SP800-56A	P-224, P-256, P-384, P-521	Shared secret establishment	Component-2042/ CAVP-72
KBKDF Component CVL	SP800-108	KDF Mode: Counter MAC Mode: HMAC-SHA-1, HMAC-SHA2-224, HMAC-SHA2-256, HMAC-SHA2-384, HMAC-SHA2-512 Supported Lengths: 1984, 2048, 4032, 4096	The output of modules' DRBG along with DH/ECDH established shared secret are used to derive symmetric key using SHA256 & HMAC. This KDF is compliant with NIST SP800_108_Counter_Mode_KDF	Component-1040/ CAVP-31437
DRBG	SP800-90A	AES256-CTR	Asymmetric key generation,	DRBG-

[3] With a min-entropy value of 0.917851, the minimum number of bits of entropy generated by the module for use in key generation is 7,519 bits per each 8192-bit request.

Algorithm	Standard	Modes & Key Sizes	Function Performed	Validation Number
			Symmetric key establishment	2282/ CAVP-15145
AES	FIPS-197	AES256-CBC	Storing CSPS	AES-5651/ CAVP-19659
AES	FIPS-197 SP800-38D	AES256-GMAC	Hardware traffic encryption	AES-3844/ CAVP-17890
RSA	FIPS 186-2 FIPS 186-4	RSA-3070: SigVer=FIPS 186-2, 186-4 Signature=PKCS 1.5 Modulo=2048, 4096 Hash=SHA256 RSA-3071: SigVer=FIPS 186-2 Signature=PKCS 1.5 Modulo=2048 Hash=SHA1	Digital signature verification of s/w bundle in Boot-ROM	RSA-3070/ CAVP-9495 RSA-3071/ CAVP-18126
SHA256	FIPS 180-4	SHA256	Digital signature verification of s/w bundle in Boot-ROM	SHS-4589/ CAVP-28338
CKG	SP800-133	Seeds used for generating symmetric & asymmetric keys are unmodified output from Module's DRBG.	Symmetric & Asymmetric Key Generation	Vendor Affirmed

Table 7: Approved Algorithms on the Module

2.9.2 Non-Approved But Allowed Algorithms

The module uses the following algorithms that are not FIPS approved but allowed

1. Diffie-Hellman: For key agreement/ establishment purposes. The supported DH groups provide minimum 112 bits of security strength.
2. Elliptic-Curve-Diffie-Hellman: For key agreement/ establishment purposes. The supported ECDH curves provide minimum 112 bits of security strength.
3. NDRNG (Non Deterministic Random Number Generator)

2.9.3 Key Zeroization

Most of the persistent keys of the module are not zeroized under normal conditions. We shall discuss the ones that are cleared through various means, under specific conditions:

1. Ephemeral keys (**HEK, HDK**) are downloaded to hardware for traffic handling. These keys in hardware are periodically replaced with new keys through a key-rotation mechanism. Additional if a service is deleted or the module is reset/ power-cycled, these keys are discarded (and new ones are established later).

2. Each ephemeral key is negotiated with remote peer by the s/w application running on the module. Once a key is established, it's programmed in the hardware and then the corresponding copy of the key from RAM is zeroed. The hardware buffers are 'write-only' i.e. the s/w may not read back the keys written.

3. The KEK (Key used to encrypt the RSA key-pair of the module) is zeroed if there is a tamper event OR upon power failure in the onboard battery. Once the KEK is destroyed, the key-pair (M_private, M_public) is effectively & permanently lost since the KEK is necessary to decrypt the stored, encrypted key-pair.

4. In addition to KEK zeroization, the module produces evidence of tampering through one or more seals and provides electronic indication (LED, console message) as well.

5. The above clauses meet FIPS level-2 requirements described in section 4.5.1 of FIPS140-2.

2.9.4 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. All the keys (persistent and ephemeral) are stored in components shielded by FIPS level-2 compliant tamper cover.

3. Redundant copies of keys are avoided. None of the keys are stored in plain-text such that these could be read back before or after a tamper event.

2.9.5 Storage of Keys

The following table covers various keys and their storage location and format.

Parameter	Type	Storage Location	Storage Format
HEK	AES256 key	Write-Only hardware	Unknown. Internal hardware implementation is not disclosed by the chip vendor.
HDK	AES256 key	Write-Only hardware	Unknown. Internal hardware implementation is not disclosed by the chip vendor.
M_private	RSA private key	Embedded flash memory	AES256 encrypted.
M_Public	RSA public key	Embedded flash memory.	AES256 encrypted.
KEK	AES256 key	Embedded RAM with a battery backup	Obfuscated form.
M_cert	X.509 certificate	Flash chip	Base64 format. Certificate is publicly shared.
CA_cert	X.509 certificate	Flash chip	Base64 format. Certificate is publicly shared.
CO_pwd	[8..19] octet text	Embedded Flash	Obfuscated form.
Boot_public	RSA public key	Embedded ROM	Base64 format. Public key can be shared with others.

Table 8: Key Storage

2.10 Self-Tests

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

2.10.1 Power-up Self Tests

As the module is powered up[4], a set of firmware and hardware tests are automatically launched. The module raises appropriate alarms and suspends its normal operation if there is a self-test failure at this stage.

The operator must either reset or power-cycle the card that has failed one or more self-tests. Subsequently the tests are initiated again. If the failures are persistent, the card may have to be replaced.

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

2.10.1.1 Pre-Boot Checks

As a part of power-sequencing the onboard CPU is held under reset. The Pre-boot check phase covers a set of tests performed by on-board custom hardware before the CPU reset is removed.

1. The on-board tamper detection circuitry checks if there was a tamper event prior to the ongoing initialization. In the case of tamper, the module halts further initialization and turns on the designated LED indicating tamper event. This step prevents erroneous/ intentional bypassing of tamper failures. A board once tampered may not be 'fixed' through a power-cycle/ reset. It must be sent back to ECI.
2. The custom logic checks if there is a valid CO password assigned to the module. The module is shipped from ECI factory without CO password. The CO must assign a valid password to the card before the initialization can proceed. The LED status changes if the password is missing.
3. The logic then checks for power failures of the on-board battery or tamper while the card was previously not powered up. LED is updated and a message is generated on the debug (console) port of the card. The CO must follow the corresponding documentation and inspect the card before giving consent to recover from this condition.
4. The custom logic releases reset on the CPU if none of the above error conditions occur. The CPU starts executing the Secure-Boot-ROM code at this point.

2.10.1.2 Secure Boot Checks

Boot-ROM is secured under the tamper shield. If the pre-boot phase passes, it implies that the boot-ROM is secure. The boot-ROM application performs the following tests:

1. Boot-ROM has a FIPS compliant implementation of cryptographic algorithms necessary for its operation. It runs self-tests for each of these algorithms as per FIPS requirements. These tests are based on the KAT approach. A failure in any of these self-tests is reflected in the form of LED pattern (which will be 'stuck' in 'boot-ROM-tests' pattern) and messages logged to the debug port. Operator / CO intervention is needed at this point.

[4] "Power Up" includes the reset operation for Apollo modules. The reset in Apollo is implemented from the System card using "chassis power reset <slot>" command which triggers power-cycle of the module.

2. After successfully running the self-tests, the boot-ROM code fetches a file containing software bundle from the System card. This file is a software bundle that contains the operating system (OS), the s/w application and various other supporting components such as FPGA images. The file is digitally signed by ECI's secure build servers at the time of its creation.
3. The boot-ROM has the necessary, in-built RSA-public key[5] to verify digital signature of the software bundle. This public key corresponds to the private-key of the secure signing server at ECI.
4. A successful signature verification of the software bundle implies that the bundle is authentic (generated by ECI) and free of tampering. Therefore, a single signature verification proves the authenticity of all the components of the software bundle.
5. After signature verification, the Boot-ROM extracts various software components from the bundle, loads these in System RAM and then transfers the execution flow to the operating system that's loaded in System RAM.
6. Effectively therefore the ROM propagates the 'chain-of-trust' to the OS, which then transfers it to the s/w application that runs as a user process of the OS. The s/w application (upon successful initialization) can then report to the management plane (i.e. System card) that it's running in secure & FIPS compliant mode.

2.10.1.3 Checks Performed by the Embedded Application

The 'chain-of-trust' is assured if the OS starts execution. The OS launches the proprietary embedded application as a user space process. Rest of the self-tests are executed by the embedded application as follows:

1. The application loads libraries related to cryptographic algorithms. It then tests each algorithm through the KAT (Known Answer Test) approach that is used[6] by the module. The answers are statically stored in the embedded application, whose integrity is guaranteed through the 'chain-of-trust' discussed above.
2. Failure in any of the self-tests leads to alarms and log messages and a different LED pattern. On the other hand, if the algorithmic self-tests are successful, the application proceeds with testing other components of the module.
3. Sanity check is performed on the TRNG (True/hardware random number generator) to ensure that it's not 'latched' to some fixed pattern.
4. The unique RSA key pair (M_private, M_public) of the module is stored in an encrypted form onboard. It's read and decrypted using the KEK stored elsewhere inside the module. Then the key-pair is checked for consistency using the KAT approach.
5. Public key from M_cert (X.509 certificate of the module) is matched with M_public.
6. Then the digital signature inside the M_cert is verified using the public key from CA_cert.
7. The CA_cert is verified next.
    a. If it's a root certificate, its digital signature is verified using the public key in the certificate itself.
    b. If CA_cert is not a root certificate, there will be a chain of certificates installed on the card. The entire chain is verified until the root certificate of that chain. Root certificates are always self-signed and verified using their own public key.

[5] As of R9.0 of Apollo we use RSA2048-bit-modulo with SHA256 for signature verification.
[6] As of R9.0 the module checks RSA, Diffie-Hellman, HMAC, AES256-CBC algorithmic implementations.

8. Integrity of the hardware encryption/ decryption channels (that are used to process customer traffic at line rate) is checked next. These channels can be programmed for SSE (Single Shot Encryption), where a single frame containing test-data can be encrypted/ decrypted using the s/w supplied parameters. Outputs of encryption & decryption blocks are matched against the known answers stored in the software application (i.e. the KAT approach).
9. Failure in any of the above tests leads to management alarms, log messages and appropriate LED indications.
10. The module enters normal operational mode once all self-tests pass. This is reflected in the status information that System card can retrieve from the module.
11. Services can be provisioned on a module that has successfully initialized. If encryption is enabled on a service instance, it negotiates symmetric keys (HEK, HDK) with the remote peer across the control plane.

2.10.1.4 Summary of KAT Tests

The following list covers various KAT (Known-Answer-Test) based tests performed by the module from various subsections described above.

Algorithm	Type of Test	Module Execution Phase
RSA verifier	KAT	Boot-ROM execution
SHA256	KAT	Boot-ROM execution
HMA-SHA256	KAT	Embedded application execution
RSA signing & verification	KAT	Embedded application execution
AES256-CBC	KAT	Embedded application execution
AES256-GMAC (in h/w)	KAT	Embedded application execution triggers this tests in h/w and verifies the results using KAT
DRBG	KAT, Health-check	Embedded application execution
ECDH	KAT	Embedded application execution
DH	KAT	Embedded application execution

Table 9: Summary of Self-Tests

2.10.2 Conditional Self Tests

The following list covers various conditional tests performed by the module:

1. RSA Pairwise Consistency Test
2. DRBG Continuous Random Number Generator Test
3. NDRNG Continuous Random Number Generator Test
4. Bypass Test

2.11 Design Assurance

Best design practices are followed & various measures are taken to assure sound firmware and hardware design of the module.

1. We use object oriented software 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 designs.

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.

2.12 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

The board will operate in FIPS approved mode under normal operating conditions. There is no special configuration option required for this. The board provides an operational information to the System card that's indicative of the FIPS compliance mode.

When a board is unable to operate in FIPS approved mode, the LEDs (specifically the 'Security LED') and the controller card (e.g. 'show chassis alarms' CLI command) show the relevant status. The Security LED of a fully initialized board, loaded with all CSPS shows solid green color. User documentation provides further details on interpreting other LED patterns corresponding to various failures/ alarms on the board.

A service instance with encryption enabled always runs in FIPS mode. If it's unable to operate securely, it doesn't transmit any client data on the line side. Each encrypted service instance has a flag indicating its FIPS status.

3.1 Module Installation

There are two different installation scenarios for a Module:

3.1.1 New Module Arriving from Factory

A new module arrives in sealed packaging. The CO must verify the packaging as well as the integrity of seals that are applied across factory installed tamper covers. These factory installed seals provide the necessary FIPS level-2 compliant protection against tamper.

Any additional seals placed by the customer as a part of installation of module in a chassis provide an additional layer of operational security that helps detect unauthorized/ unplanned movement of the module.

A new module is shipped without any CO-password. Once powered, initialization is halted for a module without a valid CO password. The CO must configure a valid password through the module's debug port. Detailed instructions regarding this are shared with customers.

3.1.2 Re-Installation in another Slot

A previously installed/ configured module may be relocated to another slot of the same or a different chassis, if needed. However, the CO must verify integrity of various tamper-indication seals before doing so. Checking the factory installed tamper seals is essential for FIPS level-2 compliance.

Previously configured CO password will remain valid until it is explicitly modified by the CO.

Once a module is installed in a slot and CO password is configured, the CO is advised to also apply optional tamper-indication seals across the front panel that will help detect unauthorized removal of the board from its slot as well any access to its debug ports. Front panel seals provide an additional layer of security even though the basic (tamper-indication) FIPS requirements are covered by the factory installed tamper seals that are applied across the module's tamper covers.

Additional tamper seals can be ordered separately (part# X93617) as each seal can only be used once.

3.2 Initial Key Loading

In addition to the physical installation & inspection of the module, the CO is also responsible for loading persistent CSPS. Subsequent to the CO password configuration, the CO should load the 'self-key-pair' (i.e. {*M_private, M_public*} discussed earlier) and the corresponding 'self-certificate' (i.e. *M_cert*). Additionally, the CO must also load a CA certificate[7] (i.e. *CA_cert*). These parameters are stored in various kinds of persistent memories of the module, which are all shielded by the tamper cover.

Subsequent to the above installation process, a module can operate in FIPS mode, as it can securely establish encryption keys with remote peers and encrypt/ decrypt client traffic using those keys. Configuration of services is handled by operators. CO's role remains limited to periodic inspection of the module to ensure that it's not tampered.

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. A properly configured module, loaded with CSPS can securely communicate with peer modules, without assuming the integrity of any intermediate network nodes (such as the System card). This is achieved through bi-directional verification of X.509 certificates between peers that are signed by a known & trusted CA.

2. The System card and the Ethernet interface between the System card and the module are trusted in so far as the service configuration/ management is concerned. The System card sends

[7] In Apollo R9.0 we only support a single CA certificate which also needs to be the root-CA. In future releases we may support a chain of CA certificates.

proprietary messages to control services on the module and collect statistics and other operational data.

3. In this context the management application (from the System card) or human operators (that are not CO) are considered as 'administrators'.

4. It should however be noted that the administrators are not allowed to modify the CSPS loaded by the CO on the System card. The CO authenticates with the module using a separate CO password (CO password is stored in a protected persistent memory of the module).

5. Administrators can initiate a '*chassis power reset*' operation to restart a module. This procedure is described in product documentation that's shared with end users [Ref: "FIPS Level-2 Products: User Guide for Cryptographic Officers, V1.0, Catalog No. X93402 "].

6. If a module is power-cycled, new keys for encrypting/ decrypting each traffic stream are established. This is part of the FIPS compliance since a combination of <key + Nonce> may not be reused by a FIPS compliant module.

3.4 Security Officer

1. There is a designed 'CO' for each module. This person is responsible for physical integrity of the module as well as in charge of loading/ updating CSPS stored onboard.

2. Additionally, the establishment of symmetric encryption keys with a remote peer is attributed to a 'CO' role in the FIPS parlance even though this operation is performed automatically by the embedded application software running on the module.

3.5 Documentation Note

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

1. Documentation for this module 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. 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 reviewed and maintained as a part of our design assurance process.

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

5. All of the above documents are controlled. Their revisions are tracked through a document management system (Sharepoint) within ECI. Each document has a unique associated ID.

6. Vendor Evidence is a supplementary document for FIPS purposes. Its primary audience is the CMVP that can get specific reference/ clarifications from ECI.

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 number visible on these parts. The module 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. The software application from the System card exports revision information from each management applications.

The module is comprised of 2 PCBs: The base-board & a mezzanine card mounted on the top of the base board. The mezzanine card functions as an adapter for the line port(s) and it's outside the tamper covers. Additionally, the pluggable transceivers (XFP or SFP) on each port contain PROM data for their identification.

There are two tamper covers fastened to the base-board. These covers have FIPS compliant tamper indicator seals placed at strategic points. Tamper seals have ECI proprietary information in encoded form.

3.6.2 Firmware Traceability

1. The boot-ROM image resident on the module has an associated revision that can be seen through an operational command executed on the System card. [ref. "*show chassis card software <slot-number>*"]
2. There is also a revision associated with the s/w bundle operational on the module which can be accessed through an operational command. [ref. "*show chassis card software <slot-number>*"]
3. The cryptographic components of the boot-ROM are versioned and tracked separately. This information can be retrieved through a boot-ROM CLI command called "*fipsrev*".

Acronyms

Keyword	Explanation
CAVP	Cryptographic Algorithm Validation Program
CCCS	Canadian Centre for Cyber Security
CLI	Command Line Interface
CMVP	Cryptographic Module Validation Program
DH	Diffie Hellman (Key establishment Algorithm)
CSP	Critical Security Parameter
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
KDF	Key Derivation Function
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
RSA	Rivest–Shamir–Adleman (asymmetric cipher algorithm)
SCM	Software Configuration Management
SW-U	Software In Use
TAC	Technical Assistance
TM200EN	Coded name of the ECI TM200EN Encryption Module. **TM** stands for Transponder Module, **200** stands for 200Gbps line-side capacity and **EN** stands for the encryption functionality.
UML	Unified Modeling Language