MMA10G-EXE Series Security Target

Document Version: 1.2
Date: 1st Oct, 2026

2400 Research Blvd
Suite 395
Rockville, MD 20850

Revision History

Version	Date	Changes
Version 0.1	24th September 2025	Initial Release
Version 0.2	30th October 2025	Updates based on GA responses – part 1
Version 0.3	6th November 2025	Updates based on GA responses – part 2
Version 0.4	23rd February 2026	Ready for Checkin
Version 0.5	27th February 2026	Checkin version without comments
Version 0.6	12th March 2026	Updates based on comments
Version 0.7	17th March 2026	Updates based on Vendor
Version 0.8	8th June 2026	Updates based on QA
Version 0.9	7th July 2026	Updates based on Check-in ECR
Version 1.0	17th September 2026	Updates based on Checkout ECR
Version 1.1	28th September 2026	Updates based on Checkout ECR
Version 1.2	01st October 2026	Updates based on internal view comments

Contents

1. Introduction 5
1.1 Security Target and TOE Reference 5
1.2 TOE Overview 5
1.3 TOE Description 6
1.3.1 Physical Boundaries 7
1.3.2 Security Functions Provided by the TOE 11
1.3.3 TOE Documentation 15
1.3.4 References 15
1.4 TOE Environment 16
1.5 Product Functionality not Included in the Scope of the Evaluation 17
2. Conformance Claims 17
2.1 CC Conformance Claims 17
2.2 Protection Profile Conformance 17
2.3 Conformance Rationale 18
2.3.1 Technical Decisions 18
3. Security Problem Definition 19
3.1 Threats 19
3.2 Assumptions 20
3.3 Organizational Security Policies 22
4. Security Objectives 23
4.1 Security Objectives for the Operational Environment 23
5. Security Requirements 25
5.1 Conventions 26
5.2 Security Functional Requirements 26
5.2.1 Security Audit (FAU) 26

5.2.2 Cryptographic Support (FCS) 30
5.2.3 Identification and Authentication (FIA) 36
5.2.4 Security Management (FMT) 38
5.2.5 Protection of the TSF (FPT) 39
5.2.6 TOE Access (FTA) 41
5.2.7 Trusted Path/Channels (FTP) 41
5.3 TOE SFR Dependencies Rationale for SFRs 42
5.4 Security Assurance Requirements 42
5.5 Assurance Measures 43
6. TOE Summary Specification 44
6.1 TOE Summary Specification SFR Description 44
6.1.1 Security Audit (FAU) 44
6.1.2 Cryptographic Support (FCS) 45
6.1.3 Identification and Authentication (FIA) 51
6.1.4 Security Management (FMT) 53
6.1.5 Protection of the TSF (FPT) 57
6.1.6 TOE Access (FTA) 59
6.1.7 Trusted Path/Channels (FTP) 60
6.2 CAVP Algorithm Certificate Details 61
6.3 Cryptographic Key Destruction 63
7. Acronym Table 65

1. INTRODUCTION

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

1.1 SECURITY TARGET AND TOE REFERENCE

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

Table 1 – TOE/ST Identification

Category	Identifier
ST Title	MMA10G-EXE Series Security Target
ST Version	1.2
ST Date	10-01-2026
ST Author	Intertek Acumen Security
TOE Identifier	MMA10G-EXE Series
TOE Version	v1.11
TOE Developer	Evertz Microsystems Ltd. 5292 John Lucas Drive Burlington, Ontario CANADA
Key Words	Network Device

1.2 TOE OVERVIEW

The MMA10G-EXE Series switches are Internet Protocol (IP) switches optimized for video-over-IP traffic (compressed or uncompressed). The TOE is classified as a network device (a generic infrastructure device that can be connected to a network). Models of the TOE included in the evaluation provide identical functionality. The only differences between them are the supported speed, the physical size, and the number of physical interfaces supported, and the processor. These differences are detailed at the end of this section.

The TOE builds on the capabilities of the existing Evertz line of video routing switches. Video routers receive video signals in various formats, such as Serial Digital Interface (SDI), Serial Data Transport Interface (SDTI), or Asynchronous Serial Interface (ASI), and switch dedicated physical input ports to dedicated physical output ports based on external commands.

The TOE provides the same capability within the context of packet-based networks using shared network infrastructure. The TOE provides a packet-based switching fabric from a video perspective, rather than relying on traditional packet-based network architecture.

A typical TOE installation will also include a standard video routing switch software platform (such as Evertz MAGNUM) to route data between program streams in a manner sufficient to meet broadcast video standards for signal availability and integrity. Equipment to prepare video for IP transport, or to convert it into other video formats, and non-network-based video switching/processing, is outside the scope of this TOE. Such equipment includes, but is not limited to, cameras, KVMs, codecs, video servers and video displays. Equipment to perform functions such as embedding audio and/or other information within the video stream is also outside the scope of this TOE. The TOE provides secure remote management using an HTTPS/TLS web interface. Administrators only may access TOE via a dedicated management workstation operating over an Out-of-Band Management (OOBM) network. Sites may close this OOBM network or may operate TOE within an existing OOBM as long as the topology is compliant with the security parameters listed below. Users and administrators may also access TOE software via direct connection using a terminal session.

The TOE generates audit logs and transmits the audit logs to a remote syslog server over an authenticated TLS channel. The TOE verifies the authenticity of software updates by verifying the digital signature prior to installing any update. The summary of the evaluated functionality provided by the TOE includes the following,

- Secure connectivity with remote audit servers and secure retention of audit logs locally
- Identification and authentication of the administrator of the TOE
- Secure remote administration of the TOE via TLS and secure Local administration of the TOE
- Secure access to the management functionality of the TOE
- Secure software updates
- Secure communication with the non-TOE 'video switch control systems' via TLS

The TOE firmware version 1.11 based on Linux 6.12 (CIP SLTS).

The TOE appliances are Ethernet switches optimized for video content.

1.3 TOE DESCRIPTION

This section provides an overview of the TOE architecture, including physical boundaries, security functions, and relevant TOE documentation and references. The item outlined in red is considered the TOE boundary for testing purposes.

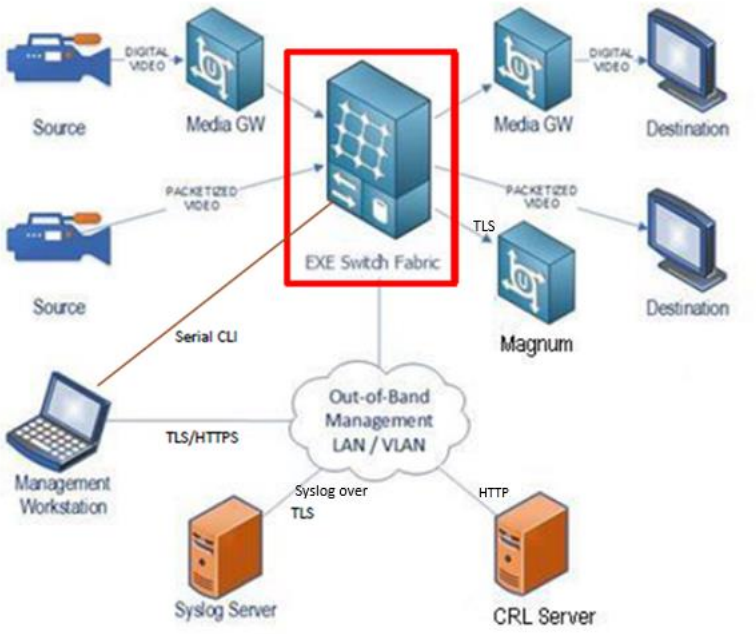


Figure 1 – Representative TOE Deployment

1.3.1 PHYSICAL BOUNDARIES

The physical boundaries of the TOE are outlined in section 1.3. The media and video components of the IT environment are NOT part of the TOE physical boundary. The TOE is shipped to the customer via commercial courier. The TOE hardware devices are the Evertz:

Table 2 TOE Models

Model	AV/ Broadcast	Supported Ports	Form Factor	Chassis Supported	Frame Controller	Processor
MMA10G-EXE16	AV	16 x QSFP28 cages per line card	16	EXE	EXE16-FC-NCS	Intel(R) Xeon(R) CPU E3-1505M v5 (Skylake)
MMA10G-EXE26	AV	16 x QSFP28 cages per line card	26	EXE	EXE-FC-NCS	Intel(R) Xeon(R) CPU E3-1505M v5 (Skylake)
MMA10G-EXE36	AV	16 x QSFP28 cages per line card	36	EXE	EXE-FC-NCS	Intel(R) Xeon(R) CPU E3-1505M v5 (Skylake)

MMA10G-EXE3.0	AV	16x DD QSFP cages per line card	9	EXE	EXE3-FC-NCS	Intel(R) Xeon(R) CPU E3-1505M v5 (Skylake)
EXE2.0-16-10G-A1	Broadcast	16 x QSFP28 cages per line card	16	EXE	EXE16-FC-NCS	Intel(R) Xeon(R) CPU E3-1505M v5 (Skylake)
EXE2.0-16-25G-A1	Broadcast	16 x QSFP28 cages per line card	16	EXE	EXE16-FC-NCS	Intel(R) Xeon(R) CPU E3-1505M v5 (Skylake)
EXE2.0-26-10G-A1	Broadcast	16 x QSFP28 cages per line card	26	EXE	EXE-FC-NCS	Intel(R) Xeon(R) CPU E3-1505M v5 (Skylake)
EXE2.0-26-25G-A1	Broadcast	16 x QSFP28 cages per line card	26	EXE	EXE-FC-NCS	Intel(R) Xeon(R) CPU E3-1505M v5 (Skylake)
EXE2.0-36-10G-A1	Broadcast	16 x QSFP28 cages per line card	36	EXE	EXE-FC-NCS	Intel(R) Xeon(R) CPU E3-1505M v5 (Skylake)
EXE2.0-36-25G-A1	Broadcast	16 x QSFP28 cages per line card	36	EXE	EXE-FC-NCS	Intel(R) Xeon(R) CPU E3-1505M v5 (Skylake)
EXE2.0-16-10G-A2	Broadcast	16 x QSFP28 cages per line card	16	EXE	EXE16-FC-NCS	Intel(R) Xeon(R) CPU E3-1505M v5 (Skylake)
EXE2.0-16-25G-A2	Broadcast	16 x QSFP28 cages per line card	16	EXE	EXE16-FC-NCS	Intel(R) Xeon(R) CPU E3-1505M v5 (Skylake)
EXE2.0-26-10G-A2	Broadcast	16 x QSFP28 cages per line card	26	EXE	EXE-FC-NCS	Intel(R) Xeon(R) CPU E3-1505M v5 (Skylake)
EXE2.0-26-25G-A2	Broadcast	16 x QSFP28 cages per line card	26	EXE	EXE-FC-NCS	Intel(R) Xeon(R) CPU E3-1505M v5(Skylake)
EXE2.0-36-10G-A2	Broadcast	16 x QSFP28 cages per line card	36	EXE	EXE-FC-NCS	Intel(R) Xeon(R) CPU E3-1505M v5 (Skylake)

EXE2.0-36-25G-A2	Broadcast	16 x QSFP28 cages per line card	36	EXE	EXE-FC-NCS	Intel(R) Xeon(R) CPU E3-1505M v5 (Skylake)
NATX-8-100G-CC	Broadcast	4 x DD QSFP (QSFP200G)	1	DragonFire frame	N/A	Intel(R) Core(TM) i3-6102E (Skylake)
NATX-16-100G-CC	Broadcast	8 x DD QSFP (QSFP200G)	1	DragonFire frame	N/A	Intel(R) Core(TM) i3-6102E (Skylake)
NATX-32-100G-1-CC	Broadcast	16 x DD QSFP (QSFP200G)	1	DragonFire frame	N/A	Intel(R) Core(TM) i3-6102E (Skylake)
NATX-64-100G-2-CC	Broadcast	32 x DD QSFP (QSFP200G)	1	DragonFire frame	N/A	Intel(R) Core(TM) i3-6102E (Skylake)
MMA10G-NATX-8-CC	AV	4 x DD QSFP (QSFP200G	1	DragonFire frame	N/A	Intel(R) Core(TM) i3-6102E (Skylake)
MMA10G-NATX-16-CC	AV	8 x DD QSFP (QSFP200G)	1	DragonFire frame	N/A	Intel(R) Core(TM) i3-6102E (Skylake)
MMA10G-NATX-32-CC	AV	16 x DD QSFP (QSFP200G)	1	DragonFire frame	N/A	Intel(R) Core(TM) i3-6102E (Skylake)
MMA10G-NATX-64-CC	AV	32 x DD QSFP (QSFP200G)	1	DragonFire frame	N/A	Intel(R) Core(TM) i3-6102E (Skylake)
MMA10G-IPX128	AV	32 x QSFP+	3 or 6	EV Frame	Ev3-FC or Ev6-FC	Intel(R) Core(TM) i3-6102E (Skylake)
3080IPX-48-25G-CC	AV/broadcast	12 x QSFP+	3 or 6	EV Frame	Ev3-FC or Ev6-FC	Intel(R) Core(TM) i3-6102E (Skylake)
MMA10G-NATX-128-CC	AV	32 x DD QSFP (QSFP200G and QSFP 400G))	1	Dragon Fire 2	N/A	Intel(R) Core(TM) i3-6102E (Skylake)

MMA10G-NATX-LT-SFP-CC	AV	32xSFP + 8x DD QSFP	1	Dragon Fire	N/A	Intel(R) Core(TM) i3-6102E (Skylake)
MMA10G-NATX-LT-QSFP-CC	AV	12xQSFP + 4x DD QSFP	1	Dragon Fire	N/A	Intel(R) Core(TM) i3-6102E (Skylake)
MMA10G-NATX-LT-MP-CC	AV	32xSFP + 8x DD QSFP	1	Dragon Fire	N/A	Intel(R) Core(TM) i3-6102E (Skylake)
NATX-LT-1G-32SFP-6QSFPDD-CC	Broadcast	32xSFP + 8x DD QSFP	1	Dragon Fire	N/A	Intel(R) Core(TM) i3-6102E (Skylake)

**Form Factor: ** The number of line-card slots the chassis can hold, indicating the system's capacity.

The Required Environmental Components used to test the TOE are shown in Table 3 below:

Table 3 – Required Environmental Components

Component	Required	Purpose/Description
Syslog server	Yes	• Conformant with RFC 5424 (Syslog Protocol) • Supporting Syslog over TLS (RFC 5425) • Acting as a TLSv1.2 server • Supporting at least one of the following cipher suites: o TLS_RSA_WITH_AES_128_CBC_SHA o TLS_RSA_WITH_AES_256_CBC_SHA o TLS_RSA_WITH_AES_128_CBC_SHA256 o TLS_RSA_WITH_AES_256_CBC_SHA256 o TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 o TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
Management Workstation with web browser	Yes	• Google Chrome, or Firefox • Supporting TLSv1.2 • Supporting Server Certificate authentication • Supporting at least one of the following ciphersuites: o TLS_RSA_WITH_AES_128_CBC_SHA o TLS_RSA_WITH_AES_256_CBC_SHA o TLS_RSA_WITH_AES_128_CBC_SHA256 o TLS_RSA_WITH_AES_256_CBC_SHA256 o TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256

Component	Required	Purpose/Description
		○ TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
Local Management Workstation	Yes	• Computer with terminal emulation software to access the console interface (CLI)
CRL Server	Yes	• Conformant with RFC 5280 • Provides a list of revoked certificates. • TOE uses the CRL server to check the revocation status of the certificates presented by server and MAGNUM video switch control system. • Communication between the TOE and the CRL server occurs over HTTP.
MAGNUM	Yes	• Provides remote management of the TOE's routing and switching of video signals. • Supporting TLSv1.2 Mutual Authentication • Supporting at least one of the following ciphersuites: ○ TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ○ TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
Media Gateway	No	• Optional component for converting media streams. Not required for TOE operations
Video Source devices	No	• Optional component for creating video streams that are sent to the TOE. Not required for TOE operations. • Supporting packetized or digital video streams.
Video Destination devices	No	• Optional component for receiving video streams that are sent from the TOE. Not required for TOE operations. • Supporting packetized or digital video streams

1.3.2 SECURITY FUNCTIONS PROVIDED BY THE TOE

The TOE provides the security functions required by the Collaborative Protection Profile for Network Devices, hereafter referred to as NDcPP v3.0e or NDcPP.

1.3.2.1 SECURITY AUDIT

The TOE's Audit security function supports audit record generation and review. The TOE provides date and time information that is used in audit timestamps. The Audit events generated by the TOE include:

- Startup and shutdown of the audit function
- Administrative login and logout events
- Changes to TSF data related to configuration changes
- Generation of a CSR and associated keypair
- Installation of a certificate
- Resetting passwords

- Failure to establish a HTTPS/TLS session
- Failure to establish a TLS session
- All use of the identification and authentication mechanism (local and remote connections to the TSF)
- Unsuccessful attempts to validate a certificate
- Initiation of a software update
- Result of a software update
- Changes to the time
- Modification of the behavior of the TSF
- Failure of self-tests
- Initiation and termination of the trusted channel
- Initiation and termination of the trusted path
- Attempts to unlock an interactive session
- Termination of a session by the session locking mechanism
- Termination of an interactive session
- Any addition, replacement or removal of trust anchors in the TOE's trust store
- Configuration of local audit settings

The TOE can store the generated audit data on itself, and it can be configured to send syslog events to a syslog server, using a TLS protected collection method. Logs are classified into various predefined categories. The logging categories help describe the content of the messages that they contain. Access to the logs is restricted to only Security Administrators, who are authorized to edit them, copy or delete (clear) them. Audit records are protected from unauthorized modifications and deletions.

The TSF provides the capability to view audit data by using the Syslog tab in the local console. The log records the time, host name, facility, application, and "message" (the log details). The previous audit records are overwritten when the allocated space for these records reaches the threshold on a FIFO basis.

1.3.2.2 CRYPTOGRAPHIC SUPPORT

The TOE includes an OpenSSL library version 3.5.4 with FIPS version 3.1.2 and Linux 6.12 (CIP SLTS) that implements CAVP validated cryptographic algorithms for random bit generation, encryption/decryption, authentication, and integrity protection/verification. These algorithms are used to provide security for the TLS/HTTPS connections for secure management and secure connections to a syslog and MAGNUM clients. TLS and HTTPS are also used to verify firmware updates. The cryptographic implementation for TOE relies on the EXE Cryptographic Module version 1.11. The cryptographic services provided by the TOE are described below:

Table 4 - TOE Cryptographic Protocols

Cryptographic Protocol	Use within the TOE
HTTPS/TLS (client)	Secure connection to syslog FCS_TLSC_EXT.1

HTTPS/TLS (server)	Peer connections to MAGNUM and remote management FCS_HTTPS_EXT.1, FCS_TLSS_EXT.1, FCS_TLSS_EXT.2
AES	Provides encryption/decryption in support of the TLS protocol. FCS_TLSC_EXT.1, FCS_TLSS_EXT.1, FCS_TLSS_EXT.2
DRBG	Deterministic random bit generation use to generate keys. FCS_TLSS_EXT.1, FCS_TLSS_EXT.2, FCS_RBG_EXT.1
Secure hash	Used as part of digital signatures and firmware integrity checks. FCS_COP.1/Hash, FCS_TLSC_EXT.1, FCS_TLSS_EXT.1, FCS_TLSS_EXT.2.
HMAC	Provides keyed hashing services in support of TLS. FCS_COP.1/KeyedHash, FCS_TLSC_EXT.1, FCS_TLSS_EXT.1, FCS_TLSS_EXT.2
EC-DH	Provides key establishment for TLS. FCS_CKM.2, FCS_TLSC_EXT.1, FCS_TLSS_EXT.1, FCS_TLSS_EXT.2
ECDSA	Provides components for EC-DH key establishment. FCS_CKM.1, FCS_CKM.2, FCS_TLSS_EXT.1, FCS_TLSS_EXT.2
RSA	Provide key establishment, key generation and signature generation and verification (PKCS1_V1.5) in support of TLS. FCS_CKM.1, FCS_COP.1/SigGen, FCS_COP.1/SigVer, FCS_TLSC_EXT.1, FCS_TLSS_EXT.1, FCS_TLSS_EXT.2

1.3.2.3 IDENTIFICATION AND AUTHENTICATION

All Administrators wanting to use TOE services are identified and authenticated prior to being allowed access to any of the services other than the display of the warning banner. ("Regular" TOE users do not access TOE directly; they control IP video switching through the TOE using a switch control system, such as Evertz's MAGNUM. The switching of those IP video transport streams is outside the scope of the TOE.)

Once an Administrator attempts to access the management functionality of the TOE, the TOE prompts the Administrator for a username and password for password-based authentication. The identification and authentication credentials are confirmed against a local user database. Only after the Administrator presents the correct identification and authentication credentials will access to the TOE functionality be granted. If the user fails to provide the correct authentication credentials, the user will be locked out after a configurable threshold until the user is manually unlocked by an Administrator.

The TOE provides the capability to set password minimum length rules. This is to ensure the use of strong passwords in attempts to protect against brute force attacks. The TOE also accepts passwords composed of a variety of characters to support complex password composition. During authentication, no indication is given of the characters composing the password.

Remote administrators are locked out after a configurable number of unsuccessful authentication attempts.

The TOE requires a password-protected serial connection to perform initial configuration of the system IP address(es). Once each address is established, administrators use IP connectivity for all further administrative actions, including configuration, operations, and monitoring. The TOE uses X.509v3 certificates as defined by RFC 5280 to support authentication for TLS/HTTPS connections.

1.3.2.4 SECURITY MANAGEMENT

The TOE provides secure administrative services for management of general TOE configuration and the security functionality provided by the TOE. All TOE administration occurs either through a secure session or a local console connection. The TOE provides the ability to perform the following actions:

- Ability to administer the TOE remotely;
- Ability to configure the access banner;
- Ability to configure the remote session inactivity time before session termination;
- Ability to update the TOE, and to verify the updates using digital signature capability prior to installing those updates;
- Ability to modify the behaviour of the transmission of audit data to an external IT entity;
- Ability to manage the cryptographic keys;
- Ability to re-enable an Administrator account;
- Ability to set the time which is used for time-stamps;
- Ability to administer the TOE locally;
- Ability to generate Certificate Signing Request (CSR) and process CA certificate response;
- Ability to configure the authentication failure parameters for FIA_AFL.1;
- Ability to manage the TOE's trust store and designate X509.v3 certificates as trust anchors;
- Ability to configure the local session inactivity time before session termination or locking;

All these management functions are restricted to Security Administrators who are authorized to administer the TOE via a local CLI and a remote web interface. Administrators are individuals who manage specific types of administrative tasks. The TOE implements role-based access control of these management functions to users that have been identified, authenticated, and authorized with the Security Administrator role.

Primary management is done using the WebEasy web-based interface using HTTPS. This provides a network administration console from which administrators can manage various identity services. These services include authentication, authorization, and reporting. All these services can be managed from the interface, which uses a menu-driven navigation system.

There is also a very simple serial-based connection (RS-232) that provides a simple menu interface. This is used to configure the IP interface (IP address, etc.). It is password-protected, and is typically only used once, for initial set-up.

1.3.2.5 PROTECTION OF THE TSF

The TOE will terminate inactive sessions after an Administrator-configurable time period. Once a session has been terminated the TOE requires the user to re-authenticate to establish a new session. The TOE provides protection of TSF data (authentication data and cryptographic keys). In addition, the TOE internally maintains the date and time. This date and time are used as the time stamp that is applied to TOE generated audit records. The TOE also ensures firmware updates are from a reliable source. Finally, the TOE performs self-testing to ensure that its cryptographic functions operate correctly and that the TOE software is running the correct image.

An administrator initiates update processes from the web interface for all update installations. TOE automatically uses the RSA digital signature mechanism to confirm the integrity of the product before installing the update.

1.3.2.6 TOE ACCESS

Aside from the automatic Administrators session termination due to inactivity described above, the TOE also allows Administrators to terminate their own interactive session. Once a session has been terminated the TOE requires the user to re-authenticate to establish a new session. The TOE will display an Administrator-specified banner on the web browser management and local console interface prior to allowing any administrative access to the TOE.

1.3.2.7 TRUSTED PATH/CHANNELS

The TOE allows the establishment of a trusted channel between a video control system (such as Evertz' MAGNUM) and the TOE. The TOE also establishes a secure connection for sending syslog data to a syslog server using TLS. The TOE uses HTTPS/TLS to provide a trusted path between itself and remote administrative users.

The TOE does not implement any additional methods of remote administration. The remote administrative users are responsible for initiating the trusted path when they wish to communicate with the TOE.

1.3.3 TOE DOCUMENTATION

The following documents are essential to understanding and controlling the TOE in the evaluated configuration:

- MMA10G-EXE Series Security Target v1.2
- MMA10G-EXE Series Security Administrative Guide Addendum for Common Criteria, version 0.4 [EXE CC Admin Guide].

1.3.4 REFERENCES

In addition to TOE documentation, the following reference may also be valuable when understanding and controlling the TOE:

- collaborative Protection Profile for Network Devices, Version 3.0 [NDcPP]

1.4 TOE ENVIRONMENT

The following environmental components are required to operate the TOE in the evaluated configuration:

Table 5 – Required Environmental Components

Component	Description
Syslog server	• Conformant with RFC 5424 (Syslog Protocol) • Supporting Syslog over TLS (RFC 5425) • Acting as a TLSv1.2 server • Supporting all of the following cipher suites: o TLS_RSA_WITH_AES_128_CBC_SHA o TLS_RSA_WITH_AES_256_CBC_SHA o TLS_RSA_WITH_AES_128_CBC_SHA256 o TLS_RSA_WITH_AES_256_CBC_SHA256 o TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 o TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
Management Workstation with web browser	• Google Chrome, or Firefox • Supporting TLSv1.2 • Supporting Server Certificate authentication • Supporting all of the following ciphersuites: o TLS_RSA_WITH_AES_128_CBC_SHA o TLS_RSA_WITH_AES_256_CBC_SHA o TLS_RSA_WITH_AES_128_CBC_SHA256 o TLS_RSA_WITH_AES_256_CBC_SHA256 o TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 o TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
Local Management Workstation	• Computer with terminal emulation software to access the console interface (CLI).
CRL Server	• Conformant with RFC 5280 • Provides a list of revoked certificates. • TOE uses the CRL server to check the revocation status of the certificates presented by server and MAGNUM video switch control system. • Communication between the TOE and the CRL server occurs over HTTP.
MAGNUM Server	• Provides remote management of the TOE's routing and switching of video signals. • Supporting TLSv1.2 Mutual Authentication • Supporting all of the following ciphersuites:

Component	Description
	○ TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ○ TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384

1.5 PRODUCT FUNCTIONALITY NOT INCLUDED IN THE SCOPE OF THE EVALUATION

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

- SNMP Traps (Alarms)
- SNMP
- VistaLINK PRO module
- Network Time Protocol (NTP) Server
- External Authentication Servers for administrator authentication (LDAP/LDAPS)
- SSH is by default disabled in CC Mode
- Rest API

These functions are outside the TOE. Alarm monitoring is the sending of SNMP traps to an alarm monitoring system (which is assigned by an Administrator).

In addition, TOE provides IP video stream switching. This IP video switching does not provide security functionality and was therefore not evaluated and is outside the scope of the TOE. The nature of video encryption and decryption is that a video stream is encrypted at the sending end and decrypted at the receiving end; since TOE is a midpoint device and therefore does not perform encryption or decryption functionality. This functionality, while present in the TOE, was not evaluated.

2. CONFORMANCE CLAIMS

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

2.1 CC CONFORMANCE CLAIMS

The TOE is conformant to the following:

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

2.2 PROTECTION PROFILE CONFORMANCE

This ST claims exact conformance to the following:

- collaborative Protection Profile for Network Devices, Version 3.0e, (CPP_ND_V3.0E)

2.3 CONFORMANCE RATIONALE

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

2.3.1 TECHNICAL DECISIONS

All NIAP TDs issued to date and applicable to CPP_ND_V3.0E have been considered. Table 6 identifies all applicable TDs.

Table 6 – Relevant Technical Decisions

Technical Decision	Applicable (Y/N)	Exclusion Rationale (if applicable)
TD1052 - Separation of Test Definitions for DTLSv1.2 and v1.3 (Renegotiation)	No	DTLS not Claimed in ST.
TD1033 – Sunset Dates for NDcPP Configurations	Yes	
TD990 - CTR_DRBG in FCS_RBG_EXT.1.2	Yes	
TD0973 – FCS_(D)TLSS_EXT.1.3 Test 2 DHE Ciphersuite Conditionality	Yes	
TD0923 - Auditable event for FAU_STG_EXT.1 in FAU_GEN.1.2	Yes	
TD0921- Addition of FIPS PUB 186-5 for RSA	Yes	
TD0900 - Clarification to Local Administrator Access in FIA_UIA_EXT.1.3	Yes	
TD0899 - Correction of Renegotiation Test for TLS 1.2	Yes	
TD0886 - FAU_STG_EXT.1 Test 6	Yes	
TD0880 - Removal of Duplicate Selection in FMT_SMF.1.1	Yes	
TD0879 - Correction of Chapter Headings in CPP_ND_V3.0E	Yes	
TD0868 - Clarification of time frames in FCS_IPSEC_EXT.1.7 and FCS_IPSEC_EXT.1.8	No	IPsec not Claimed in ST.
TD0836 - Redundant Requirements in FPT_TST_EXT.1	Yes	

3. SECURITY PROBLEM DEFINITION

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

3.1 THREATS

The threats included in Table 7 are drawn directly from the PP specified in Section 2.2.

Table 7 – Threats

ID	Threat
T.UNAUTHORIZED_ADMINISTRATOR_ACCESS	Threat agents may attempt to gain Administrator access to the Network Device by nefarious means such as masquerading as an Administrator to the device, masquerading as the device to an Administrator, replaying an administrative session (in its entirety, or selected portions), or performing man-in-the-middle attacks, which would provide access to the administrative session, or sessions between Network Devices. Successfully gaining Administrator access allows malicious actions that compromise the security functionality of the device and the network on which it resides.
T.WEAK_CRYPTOGRAPHY	Threat agents may exploit weak cryptographic algorithms or perform a cryptographic exhaust against the key space. Poorly chosen encryption algorithms, modes, and key sizes will allow attackers to compromise the algorithms, or brute force exhaust the key space and give them unauthorized access allowing them to read, manipulate and/or control the traffic with minimal effort.
T.UNTRUSTED_COMMUNICATION_CHANNELS	Threat agents may attempt to target Network Devices that do not use standardized secure tunnelling protocols to protect the critical network traffic. Attackers may take advantage of poorly designed protocols or poor key management to successfully perform man-in-the-middle attacks, replay attacks, etc. Successful attacks will result in loss of confidentiality and integrity of the critical network traffic, and potentially could lead to a compromise of the Network Device itself.
T.WEAK_AUTHENTICATION_ENDPOINTS	Threat agents may take advantage of secure protocols that use weak methods to authenticate the endpoints, e.g. a shared password that is guessable or transported as plaintext. The consequences are the same as a poorly designed protocol, the attacker could masquerade as the Administrator or another device, and the attacker could insert themselves into the network stream and perform a man-in-the-middle attack. The result is the critical network

ID	Threat
	traffic is exposed and there could be a loss of confidentiality and integrity, and potentially the Network Device itself could be compromised.
T.UPDATE_COMPROMISE	Threat agents may attempt to provide a compromised update of the software or firmware which undermines the security functionality of the device. Non-validated updates or updates validated using non-secure or weak cryptography leave the update firmware vulnerable to surreptitious alteration.
T.UNDETECTED_ACTIVITY	Threat agents may attempt to access, change, and/or modify the security functionality of the Network Device without Administrator awareness. This could result in the attacker finding an avenue (e.g., misconfiguration, flaw in the product) to compromise the device and the Administrator would have no knowledge that the device has been compromised.
T.SECURITY_FUNCTIONALITY_COMPROMISE	Threat agents may compromise credentials and device data enabling continued access to the Network Device and its critical data. The compromise of credentials includes replacing existing credentials with an attacker's credentials, modifying existing credentials, or obtaining the Administrator or device credentials for use by the attacker. Threat agents may also be able to take advantage of weak administrative passwords to gain privileged access to the device.
T.SECURITY_FUNCTIONALITY_FAILURE	An external, unauthorized entity could make use of failed or compromised security functionality and might therefore subsequently use or abuse security functions without prior authentication to access, change or modify device data, critical network traffic or security functionality of the device.

3.2 ASSUMPTIONS

The assumptions included in Table 8 are drawn directly from PP.

Table 8 – Assumptions

A.PHYSICAL_PROTECTION	The Network Device is assumed to be physically protected in its operational environment and not subject to physical attacks that compromise the security or interfere with the device's physical interconnections and correct operation. This protection is assumed to be sufficient to protect the device and the data it contains. As a result, the cPP does not include any requirements on physical tamper protection or other physical attack mitigations. The cPP does not expect the product to defend against physical access to the device that allows unauthorized entities to extract data, bypass other controls, or otherwise manipulate the device. For vNDs, this assumption applies to the physical platform on which the VM runs.
A.LIMITED_FUNCTIONALITY	The device is assumed to provide networking functionality as its core function and not provide functionality/services that could be deemed as general purpose computing. For example, the device should not provide a computing platform for general purpose applications (unrelated to networking functionality).
A.NO_THRU_TRAFFIC_PROTECTION	A standard/generic Network Device does not provide any assurance regarding the protection of traffic that traverses it. The intent is for the Network Device to protect data that originates on or is destined to the device itself, to include administrative data and audit data. Traffic that is traversing the Network Device, destined for another network entity, is not covered by the ND cPP. It is assumed that this protection will be covered by cPPs and PP-Modules for particular types of Network Devices (e.g., firewall).
A.TRUSTED_ADMINISTRATOR	The Security Administrator(s) for the Network Device are assumed to be trusted and to act in the best interest of security for the organization. This includes appropriately trained, following policy, and adhering to guidance documentation. Administrators are trusted to ensure passwords/credentials have sufficient strength and entropy and to lack malicious intent when administering the device. The Network Device is not expected to be capable of defending against a malicious Administrator that actively works to bypass or compromise the security of the device. For TOEs supporting X.509v3 certificate-based authentication, the Security Administrator(s) are expected to fully validate (e.g. offline verification) any CA certificate (root CA certificate or intermediate CA certificate) loaded

	into the TOE's trust store (aka 'root store', ' trusted CA Ke Store', or similar) as a trust anchor prior to use (e.g. offlin verification).
A.REGULAR_UPDATES	The Network Device firmware and software is assumed to be updated by an Administrator on a regular basis in response to the release of product updates due to known vulnerabilities.
A.ADMIN_CREDENTIALS_SECURE	The Administrator's credentials (private key) used to access the Network Device are protected by the platform on which they reside.
A.RESIDUAL_INFORMATION	The Administrator must ensure that there is no unauthorized access possible for sensitive residual information (e.g. cryptographic keys, keying material, PINs, passwords etc.) on networking equipment when the equipment is discarded or removed from its operational environment.

3.3 ORGANIZATIONAL SECURITY POLICIES

The OSPs included in Table 9 are drawn directly from the PP.

Table 9 – OSPs

ID	OSP
P.ACCESS_BANNER	The TOE shall display an initial banner describing restrictions of use, legal agreements, or any other appropriate information to which Administrators consent by accessing the TOE.

4. SECURITY OBJECTIVES

The security objectives have been taken directly from the claimed PP and are reproduced here for the convenience of the reader.

4.1 SECURITY OBJECTIVES FOR THE OPERATIONAL ENVIRONMENT

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

Table 10 – Security Objectives for the Operational Environment

ID	Objectives for the Operational Environment
OE.PHYSICAL	Physical security, commensurate with the value of the TOE and the data it contains, is provided by the environment.
OE.NO_GENERAL_PURPOSE	There are no general-purpose computing capabilities (e.g., compilers or user applications) available on the TOE, other than those services necessary for the operation, administration and support of the TOE. Note: For vNDs the TOE includes only the contents of the its own VM, and does not include other VMs or the VS.
OE.NO_THRU_TRAFFIC_PROTECTION	The TOE does not provide any protection of traffic that traverses it. It is assumed that protection of this traffic will be covered by other security and assurance measures in the operational environment.
OE.TRUSTED_ADMIN	Security Administrators are trusted to follow and apply all guidance documentation in a trusted manner. For vNDs, this includes the VS Administrator responsible for configuring the VMs that implement ND functionality. For TOEs supporting X.509v3 certificate-based authentication, the Security Administrator(s) are assumed to monitor the revocation status of all certificates in the TOE's trust store and to remove any certificate from the TOE's trust store in case such certificate can no longer be trusted.
OE.UPDATES	The TOE firmware and software is updated by an Administrator on a regular basis in response to the release of product updates due to known vulnerabilities.
OE.ADMIN_CREDENTIALS_SECURE	The Administrator's credentials (private key) used to access the TOE must be protected on any other platform on which they reside.

ID	Objectives for the Operational Environment
OE.RESIDUAL_INFORMATION	The Security Administrator ensures that there is no unauthorized access possible for sensitive residual information (e.g. cryptographic keys, keying material, PINs, passwords etc.) on networking equipment when the equipment is discarded or removed from its operational environment. For vNDs, this applies when the physical platform on which the VM runs is removed from its operational environment.

5. SECURITY REQUIREMENTS

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

Table 11 – SFRs

Requirement	Description
FAU_GEN.1	Audit Data Generation
FAU_GEN.2	User Identity Association
FAU_STG_EXT.1	Protected Audit Event Storage
FCS_CKM.1	Cryptographic Key Generation
FCS_CKM.2	Cryptographic Key Establishment
FCS_CKM.4	Cryptographic Key Destruction
FCS_COP.1/DataEncryption	Cryptographic Operation (AES Data Encryption/Decryption)
FCS_COP.1/SigGen	Cryptographic Operation (Signature Generation and Verification)
FCS_COP.1/Hash	Cryptographic Operation (Hash Algorithm)
FCS_COP.1/KeyedHash	Cryptographic Operation (Keyed Hash Algorithm)
FCS_HTTPS_EXT.1	HTTPS Protocol
FCS_RBG_EXT.1	Random Bit Generation
FCS_TLSC_EXT.1	TLS Client Protocol
FCS_TLSS_EXT.1	TLS Server Protocol
FCS_TLSS_EXT.2	TLS Server Support for Mutual Authentication
FIA_AFL.1	Authentication Failure Handling
FIA_PMG_EXT.1	Password Management
FIA_UIA_EXT.1	User Identification and Authentication
FIA_UAU.7	Protected Authentication Feedback (Refinement)
FIA_X509_EXT.1/Rev	X.509 Certificate Validation
FIA_X509_EXT.2	X.509 Certificate Authentication
FIA_X509_EXT.3	X.509 Certificate Requests
FMT_MOF.1/Functions	Management of Security Functions Behaviour
FMT_MOF.1/ManualUpdate	Management of Security Functions Behaviour
FMT_MTD.1/CoreData	Management of TSF Data
FMT_MTD.1/CryptoKeys	Management of TSF Data
FMT_SMF.1	Specification of Management Functions
FMT_SMR.2	Restrictions on Security Roles
FPT_SKP_EXT.1	Protection of TSF Data (for reading of all symmetric keys)
FPT_APW_EXT.1	Protection of Administrator Passwords

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

5.1 CONVENTIONS

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

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

5.2 SECURITY FUNCTIONAL REQUIREMENTS

This section includes the security functional requirements for this ST.

5.2.1 SECURITY AUDIT (FAU)

5.2.1.1 FAU_GEN.1 AUDIT DATA GENERATION

FAU_GEN.1.1

The TSF shall be able to generate an audit record of the following auditable events:

a) Start-up and shut-down of the audit functions;
b) Auditable events for the not specified level of audit; and
c) *All administrative actions comprising:*
    - *Administrative login and logout (name of Administrator account shall be logged if individual accounts are required for Administrators).*
    - *Changes to TSF data related to configuration changes (in addition to the information that a change occurred it shall be logged what has been changed).*

- *Generating/import of, changing, or deleting of cryptographic keys (in addition to the action itself a unique key name or key reference shall be logged).*
- [Resetting passwords (name of related Administrator account shall be logged)];

d) Specifically defined auditable events listed in Table 12.

FAU_GEN.1.2

The TSF shall record within each audit record at least the following information:

a) Date and time of the event, type of event, subject identity ~~(if applicable),~~ and the outcome (success or failure) of the event; and

b) For each audit event type, based on the auditable event definitions of the functional components included in the cPP/ST, *information specified in column three of Table 12.*

Table 12 – Security Functional Requirements and Auditable Events

Requirement	Auditable Events	Additional Audit Record Contents
FAU_GEN.1	None.	None.
FAU_GEN.2	None.	None.
FAU_STG_EXT.1	Configuration of local audit settings.	Identity of account making changes to the audit configuration.
FCS_CKM.1	None.	None.
FCS_CKM.2	None.	None.
FCS_CKM.4	None.	None.
FCS_COP.1/DataEncryption	None.	None.
FCS_COP.1/SigGen	None.	None.
FCS_COP.1/Hash	None.	None.
FCS_COP.1/KeyedHash	None.	None.
FCS_HTTPS_EXT.1	Failure to establish a HTTPS Session.	Reason for failure
FCS_RBG_EXT.1	None.	None.
FCS_TLSC_EXT.1	Failure to establish a TLS Session	Reason for failure
FCS_TLSS_EXT.1	Failure to establish a TLS Session	Reason for failure
FCS_TLSS_EXT.2	Failure to authenticate the client	Reason for failure
FIA_AFL.1	Unsuccessful login attempts limit is met or exceeded.	Origin of the attempt (e.g., IP address).
FIA_PMG_EXT.1	None.	None.
FIA_UIA_EXT.1	All use of identification and authentication mechanisms.	Origin of the attempt (e.g., IP address).
FIA_UAU.7	None.	None.

Requirement	Auditable Events	Additional Audit Record Contents
FIA_X509_EXT.1/Rev	• Unsuccessful attempt to validate a certificate • Any addition, replacement or removal of trust anchors in the TOE's trust store	• Reason for failure of certificate validation • Identification of certificates added, replaced or removed as trust anchor in the TOE's trust store
FIA_X509_EXT.2	None	None
FIA_X509_EXT.3	None.	None.
FMT_MOF.1/Functions	None.	None.
FMT_MOF.1/ManualUpdate	Any attempt to initiate a manual update	None.
FMT_MTD.1/CoreData	None.	None.
FMT_MTD.1/CryptoKeys	None.	None.
FMT_SMF.1	All management activities of TSF data.	None.
FMT_SMR.2	None.	None.
FPT_APW_EXT.1	None.	None.
FPT_SKP_EXT.1	None.	None.
FPT_TST_EXT.1	None.	None.
FPT_STM_EXT.1	Discontinuous changes to time - either Administrator actuated or changed via an automated process. (Note that no continuous changes to time need to be logged. See also application note on FPT_STM_EXT.1)	For discontinuous changes to time: The old and new values for the time. Origin of the attempt to change time for success and failure (e.g., IP address).
FPT_TUD_EXT.1	Initiation of update; result of the update attempt (success or failure)	None.
FTA_SSL.3	The termination of a remote session by the session locking mechanism.	None.
FTA_SSL.4	The termination of an interactive session.	None.
FTA_SSL_EXT.1 (if "terminate the session" is selected)	The termination of a local session by the session lock.	None.
FTA_TAB.1	None.	None.

Requirement	Auditable Events	Additional Audit Record Contents
FTP_ITC.1	• Initiation of the trusted channel. • Termination of the trusted channel. • Failure of the trusted channel functions.	• None • None • Reason for failure
FTP_TRP.1/Admin	• Initiation of the trusted path. • Termination of the trusted path. • Failure of the trusted path functions.	• None • None • Reason for failure

5.2.1.2 FAU_GEN.2 USER IDENTITY ASSOCIATION

FAU_GEN.2.1
For audit events resulting from actions of identified users, the TSF shall be able to associate each auditable event with the identity of the user that caused the event.

5.2.1.3 FAU_STG_EXT.1 PROTECTED AUDIT EVENT STORAGE

FAU_STG_EXT.1.1
The TSF shall be able to transmit the generated audit data to an external IT entity using a trusted channel according to FTP_ITC.1.

FAU_STG_EXT.1.2
The TSF shall be able to store generated audit data on the TOE itself. In addition [

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

FAU_STG_EXT.1.3
The TSF shall maintain a [log file] of audit records in the event that an interruption of communication with the remote audit server occurs.

FAU_STG_EXT.1.4
The TSF shall be able to store [persistent] audit records locally with a minimum storage size of [*880MB*].

FAU_STG_EXT.1.5
The TSF shall [overwrite previous audit records according to the following rule: *[on a circular (FIFO) basis]] when* the local storage space for audit data is full.

FAU_STG_EXT.1.6
The TSF shall provide the following mechanisms for administrative access to locally stored audit records [ability to view locally, manual export].

5.2.2 CRYPTOGRAPHIC SUPPORT (FCS)

5.2.2.1 FCS_CKM.1 CRYPTOGRAPHIC KEY GENERATION

FCS_CKM.1.1 [TD0921]
The TSF shall generate **asymmetric** cryptographic key in accordance with a specified cryptographic key generation algorithm: [

- RSA schemes using cryptographic key sizes of [2048-bit] that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.3 or FIPS PUB 186-5, "Digital Signature Standard (DSS)", A.1;
- ECC schemes using 'NIST curves' [P-256, P-384, P-521] that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4, or FIPS PUB 186-5, "Digital Signature Standard (DSS)", Appendix A.2, or ISO/IEC 14888-3, "IT Security techniques - Digital signatures with appendix - Part 3: Discrete logarithm based mechanisms", Section 6.6.;

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

5.2.2.2 FCS_CKM.2 CRYPTOGRAPHIC KEY ESTABLISHMENT

FCS_CKM.2.1
The TSF shall **perform** cryptographic **key establishment** in accordance with a specified cryptographic key **establishment** method: [

- RSA-based key establishment schemes that meet the following: RSAES-PKCS1-v1_5 as specified in Section 7.2 of RFC 8017, "Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.2";
- Elliptic curve-based key establishment schemes that meet the following: NIST Special Publication 800-56A Revision 3, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography";

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

5.2.2.3 FCS_CKM.4 CRYPTOGRAPHIC KEY DESTRUCTION

FCS_CKM.4.1
The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method

- *For plaintext keys in volatile storage, the destruction shall be executed by a* [single overwrite consisting of [zeroes]];
- *For plaintext keys in non-volatile storage, the destruction shall be executed by the invocation of an interface provided by a part of the TSF that* [

- logically addresses the storage location of the key and performs a [single] overwrite consisting of [zeroes]];

that meets the following: *No Standard.*

5.2.2.4 FCS_COP.1/DATAENCRYPTION CRYPTOGRAPHIC OPERATIONS (AES DATA ENCRYPTION/DECRYPTION)

FCS_COP.1.1/DataEncryption
The TSF shall perform *encryption/decryption* in accordance with a specified cryptographic algorithm *AES used in* [CBC, CTR, GCM] *mode* and cryptographic key sizes [128 bits, 256 bits] that meet the following: *AES as specified in ISO 18033-3,* [CBC as specified in ISO 10116, CTR as specified in ISO 10116, GCM as specified in ISO 19772].

5.2.2.5 FCS_COP.1/SIGGEN CRYPTOGRAPHIC OPERATION (SIGNATURE GENERATION AND VERIFICATION)

FCS_COP.1.1/SigGen [TD0921]
The TSF shall perform *cryptographic signature services (generation and verification)* in accordance with a specified cryptographic algorithm [

- RSA Digital Signature Algorithm,

]
and cryptographic key sizes [

- For RSA: [*modulus 2048 bits*],

]
that meets the following: [

- For RSA schemes: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 5.5, using PKCS #1 v2.1 or FIPS PUB 186-5, "Digital Signature Standard (DSS)", Section 5.4 using PKCS #1 v2.2 Signature Schemes RSASSA-PSS and/or RSASSA-PKCS1v1_5; ISO/IEC 9796-2, Digital signature scheme 2 or Digital Signature scheme 3,

].

5.2.2.6 FCS_COP.1/HASH CRYPTOGRAPHIC OPERATIONS (HASH ALGORITHM)

FCS_COP.1.1/Hash
The TSF shall perform *cryptographic hashing services* in accordance with a specified cryptographic algorithm [SHA-1, SHA-256, SHA-384, SHA-512] ~~and cryptographic key sizes~~ [~~*assignment: cryptographic key sizes*~~] and **message digest sizes [*160, 256, 384, 512*] bits** that meet the following: *ISO/IEC 10118-3:2004*.

5.2.2.7 FCS_COP.1/KEYEDHASH CRYPTOGRAPHIC OPERATION (KEYED HASH ALGORITHM)

FCS_COP.1.1/KeyedHash
The TSF shall perform *keyed-hash message authentication* in accordance with a specified cryptographic algorithm [HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384] and cryptographic key sizes *[160 bits, 256 bits, and 384 bits used in*

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

5.2.2.8 FCS_HTTPS_EXT.1 HTTPS PROTOCOL

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

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

5.2.2.9 FCS_RBG_EXT.1 RANDOM BIT GENERATION

FCS_RBG_EXT.1.1
The TSF shall perform all deterministic random bit generation services in accordance with ISO/IEC 18031:2011 using [CTR_DRBG (AES)].

FCS_RBG_EXT.1.2 [TD990]
The deterministic RBG shall be seeded by at least one entropy source that accumulates entropy from *[1]* software-based noise source] with a minimum of [256 bits] of entropy at least equal to

[

- the greatest security strength according to ISO/IEC 18031:2011 Table C.2 — "Security strengths, Entropy and Seed length requirements for the AES-128, 192 and 256 Block Cipher",

].

5.2.2.10 FCS_TLSC_EXT.1 TLS CLIENT PROTOCOL

FCS_TLSC_EXT.1.1
The TSF shall implement [TLS 1.2 (RFC 5246)] supporting the following ciphersuites: [

- *Select supported ciphersuites for TLS 1.2 from List 1*
  [
  - TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268
  - TLS_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268
  - TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246
  - TLS_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246
  - TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
  - TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289

  ]

] and no other ciphersuites.

Application Note: The TOE supports non-mutual authentication in the role of a client when establishing connections to external syslog server (Audit Server).

FCS_TLSC_EXT.1.2
The TSF shall verify that the presented identifier matches [the reference identifier per RFC 6125 Section 6].

FCS_TLSC_EXT.1.3
The TSF shall not establish a trusted channel if the server certificate is invalid [

- without any administrator override mechanism.

].

FCS_TLSC_EXT.1.4
The TSF shall [present the Supported Groups Extension with the following curves/groups: [secp256r1, secp384r1, secp521r1] and no other curves/groups] in the Client Hello.

FCS_TLSC_EXT.1.5
The TSF shall [

- present the signature_algorithms extension with support for the following algorithms:
  [
  - rsa_pkcs1 with sha256(0x0401),
  - rsa_pkcs1with sha384(0x0501),
  - rsa_pkcs1 with sha512(0x0601),
  - rsa_pss_rsae with sha256(0x0804),
  - rsa_pss_rsae with sha384(0x0805),
  - rsa_pss_rsae with sha512(0x0806),

  ] and no other algorithms;

]

FCS_TLSC_EXT.1.6
The TSF [does not provide] the ability to configure the list of supported ciphersuites as defined in FCS_TLSC_EXT.1.1.

FCS_TLSC_EXT.1.7
The TSF shall prohibit the use of the following extensions:

- Early data extension
- Post-handshake client authentication according to RFC 8446, Section 4.2.6.

FCS_TLSC_EXT.1.8
The TSF shall [not use PSKs].

FCS_TLSC_EXT.1.9
The TSF shall [reject [TLS 1.2] renegotiation attempts].

5.2.2.11 FCS_TLSS_EXT.1/WEBGUI TLS SERVER PROTOCOL

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

[

- TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268
- TLS_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268
- TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246
- TLS_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289

] and no other ciphersuites.

Application Note: The TOE supports non-mutual authentication in the role of a server when providing remote administrative access via its web interface.

FCS_TLSS_EXT.1.2/WEBGUI
The TSF shall authenticate itself using X.509 certificate(s) using [RSA with key size [2048] bits].

FCS_TLSS_EXT.1.3/WEBGUI
The TSF shall perform key exchange using: [

- RSA key establishment with key size [2048] bits;
- EC Diffie-Hellman key agreement over NIST curves [secp256r1, secp384r1, secp521r1] and no other curves;

].

FCS_TLSS_EXT.1.4/WEBGUI
The TSF shall support [no session resumption].

FCS_TLSS_EXT.1.5/WEBGUI
The TSF [does not provide] the ability to configure the list of supported ciphersuites as defined in FCS_TLSS_EXT.1.1.

FCS_TLSS_EXT.1.6/WEBGUI
The TSF shall prohibit the use of the following extensions:

- Early data extension

FCS_TLSS_EXT.1.7/WEBGUI
The TSF shall [not use PSKs].

FCS_TLSS_EXT.1.8/WEBGUI
The TSF shall [reject [TLS 1.2] renegotiation attempts].

5.2.2.12 FCS_TLSS_EXT.1/MAGNUM TLS SERVER PROTOCOL

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

- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289

] and no other ciphersuites.

Application Note: The TOE supports mutual authentication in the role of a server when establishing connections to MAGNUM (Video Control Server). The MAGNUM initiates the trusted channel with the TOE. Also,

FCS_TLSS_EXT.1.2/MAGNUM
The TSF shall authenticate itself using X.509 certificate(s) using [RSA with key size [2048] bits].

FCS_TLSS_EXT.1.3/MAGNUM
The TSF shall perform key exchange using: [

- EC Diffie-Hellman key agreement over NIST curves [secp256r1, secp384r1, secp521r1] and no other curves;

].

FCS_TLSS_EXT.1.4/MAGNUM
The TSF shall support [no session resumption].

FCS_TLSS_EXT.1.5/MAGNUM
The TSF [does not provide] the ability to configure the list of supported ciphersuites as defined in FCS_TLSS_EXT.1.1.

FCS_TLSS_EXT.1.6/MAGNUM
The TSF shall prohibit the use of the following extensions:

- Early data extension

FCS_TLSS_EXT.1.7/MAGNUM
The TSF shall [not use PSKs].

FCS_TLSS_EXT.1.8/MAGNUM
The TSF shall [reject [TLS 1.2] renegotiation attempts].

5.2.2.13 FCS_TLSS_EXT.2 TLS SERVER SUPPORT FOR MUTUAL AUTHENTICATION

FCS_TLSS_EXT.2.1
The TSF shall support TLS communication with mutual authentication of TLS clients using X.509v3 certificates and shall [

- reject the connection if the client either does not provide a client certificate at all or the client certificate cannot be successfully validated by the TOE (except for override mechanisms that might be defined in FCS_TLSS_EXT.2.2) ('hard fail')

].

Application Note: The TOE supports mutual authentication in the role of a server when establishing connections to MAGNUM (Video Control Server). The MAGNUM initiates the trusted channel with the TOE. The WEBGUI does not support mutual authentication.

FCS_TLSS_EXT.2.2
When establishing a trusted channel, by default the TSF shall not establish a trusted channel if the client certificate is invalid. The TSF shall also [

- not implement any administrator override mechanism

].

FCS_TLSS_EXT.2.3
The TSF shall not establish a trusted channel if the identifier contained in a certificate does not match an expected identifier for the client. If the identifier is a Fully Qualified Domain Name (FQDN), then the TSF shall match the identifiers according to RFC 6125, otherwise the TSF shall parse the identifier from the certificate and match the identifier against the expected identifier of the client as described in the TSS.

FCS_TLSS_EXT.2.4
The TSF shall present a [TLS 1.2] Certificate Request message containing the following algorithms: [

- rsa_pkcs1 with sha256(0x0401),
- rsa_pkcs1 with sha384(0x0501),
- rsa_pkcs1 with sha512(0x0601),
- rsa_pss_rsae with sha256(0x0804),
- rsa_pss_rsae with sha384(0x0805),
- rsa_pss_rsae with sha512(0x0806),

] and no other algorithms.

5.2.3 IDENTIFICATION AND AUTHENTICATION (FIA)

5.2.3.1 FIA_AFL.1 AUTHENTICATION FAILURE HANDLING

FIA_AFL.1.1
The TSF shall detect when an Administrator configurable positive integer within [*3 to 20*] unsuccessful authentication attempts occur related to *Administrators attempting to authenticate remotely using a password*.

FIA_AFL.1.2
When the defined number of unsuccessful authentication attempts has been met, the TSF shall [prevent the offending Administrator from successfully establishing a remote session using any authentication method that involves a password until [*an Administrator unlocks the user*] is taken by an Administrator; prevent the offending Administrator from successfully establishing a remote session using any authentication method that involves a password until an Administrator defined time period has elapsed].

5.2.3.2 FIA_PMG_EXT.1 PASSWORD MANAGEMENT

FIA_PMG_EXT.1.1
The TSF shall provide the following password management capabilities for administrative passwords:

a) Passwords shall be able to be composed of any combination of upper and lower case letters, numbers and the following special characters: [*"!", "@", "#", "$", "%", "^", "&", "*", "(", ")", ["~", "`", " ", "-", "+", "=", "{", "[", "}", "]", "|", "\", ":", ";", (''), (’), "<", ",", ">", ".", "?", "/", (space)*]] ;

b) Minimum password length shall be *configurable to between [15] and [20] characters.*

5.2.3.3 FIA_UIA_EXT.1 USER IDENTIFICATION AND AUTHENTICATION

FIA_UIA_EXT.1.1
The TSF shall allow the following actions prior to requiring the non-TOE entity to initiate the identification and authentication process:

- Display the warning banner in accordance with FTA_TAB.1;
- [no other actions].

FIA_UIA_EXT.1.2
The TSF shall require each administrative user to be successfully identified and authenticated before allowing any other TSF-mediated actions on behalf of that administrative user.

FIA_UIA_EXT.1.3 [TD0900]
The TSF shall provide the following remote authentication mechanisms [Web GUI password] and [no other mechanism]. The TSF shall provide the following local authentication mechanisms [password-based].

FIA_UIA_EXT.1.4
The TSF shall authenticate any administrative user's claimed identity according to each authentication mechanism specified in FIA_UIA_EXT.1.3.

5.2.3.4 FIA_UAU.7 PROTECTED AUTHENTICATION FEEDBACK (REFINEMENT)

FIA_UAU.7.1
The TSF shall provide only *obscured feedback* to the **administrative** user while the authentication is in progress **at the local console**.

5.2.3.5 FIA_X509_EXT.1/REV X.509 CERTIFICATE VALIDATION

FIA_X509_EXT.1.1/Rev
The TSF shall validate certificates in accordance with the following rules:

- RFC 5280 certificate validation and certification path validation **supporting a minimum path length of three certificates**.
- The certification path must terminate with a trusted CA certificate designated as a trust anchor.
- The TSF shall validate a certification path by ensuring that all CA certificates in the certification path contain the basicConstraints extension with the CA flag set to TRUE.
- The TSF shall validate the revocation status of the certificate using [a Certificate Revocation List (CRL) as specified in RFC 5280 Section 6.3].
- The TSF shall validate the extendedKeyUsage field according to the following rules:
  - *Certificates used for trusted updates and executable code integrity verification shall have the Code Signing purpose (id-kp 3 with OID 1.3.6.1.5.5.7.3.3) in the extendedKeyUsage field.*
  - *Server certificates presented for DTLS/TLS shall have the Server Authentication purpose(id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field.*

- *Client certificates presented for DTLS/TLS shall have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the extendedKeyUsagefield.*
- *OCSP certificates presented for OCSP responses shall have the OCSP Signing purpose(id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the extendedKeyUsage field.*

FIA_X509_EXT.1.2/Rev
The TSF shall only treat a certificate as a CA certificate if the basicConstraints extension is present and the CA flag is set to TRUE.

5.2.3.6 FIA_X509_EXT.2 X.509 CERTIFICATE AUTHENTICATION

FIA_X509_EXT.2.1
The TSF shall use X.509v3 certificates as defined by RFC 5280 to support authentication for [HTTPS, TLS] and [no additional uses].

FIA_X509_EXT.2.2
When the TSF cannot establish a connection to determine the validity of a certificate, the TSF shall [not accept the certificate].

5.2.3.7 FIA_X509_EXT.3 X.509 CERTIFICATE REQUESTS

FIA_X509_EXT.3.1
The TSF shall generate a Certificate Request as specified by RFC 2986 and be able to provide the following information in the request: public key and [Common Name, Organization, Organizational Unit, Country].

FIA_X509_EXT.3.2
The TSF shall validate the chain of certificates from the Root CA upon receiving the CA Certificate Response.

5.2.4 SECURITY MANAGEMENT (FMT)

5.2.4.1 FMT_MOF.1/FUNCTIONS MANAGEMENT OF SECURITY FUNCTIONS BEHAVIOUR

FMT_MOF.1.1/Functions
The TSF shall restrict the ability to [modify the behaviour of] the functions [transmission of audit data to an external IT entity] to *Security Administrators*.

5.2.4.2 FMT_MOF.1/MANUALUPDATE MANAGEMENT OF SECURITY FUNCTIONS BEHAVIOR

FMT_MOF.1.1/ManualUpdate
The TSF shall restrict the ability to enable the functions *to perform manual updates* to *Security Administrators.*

5.2.4.3 FMT_MTD.1/COREDATA MANAGEMENT OF TSF DATA

FMT_MTD.1.1/CoreData
The TSF shall restrict the ability to *manage* the *TSF data* to *Security Administrators*.

5.2.4.4 FMT_MTD.1/CRYPTOKEYS MANAGEMENT OF TSF DATA

FMT_MTD.1.1/CryptoKeys
The TSF shall restrict the ability to *manage* the *cryptographic keys* to *Security Administrators*.

5.2.4.5 FMT_SMF.1 SPECIFICATION OF MANAGEMENT FUNCTIONS

FMT_SMF.1.1 [TD0880]
The TSF shall be capable of performing the following management functions:

- *Ability to administer the TOE remotely;*
- *Ability to configure the access banner;*
- *Ability to configure the remote session inactivity time before session termination;*
- *Ability to update the TOE, and to verify the updates using* [digital signature] *capability prior to installing those updates;*

[

  - Ability to modify the behaviour of the transmission of audit data to an external IT entity;
  - Ability to manage the cryptographic keys;
  - Ability to re-enable an Administrator account;
  - Ability to set the time which is used for time-stamps;
  - Ability to manage the TOE's trust store and designate X509.v3 certificates as trust anchors;
  - Ability to generate Certificate Signing Request (CSR) and process CA certificate response;
  - Ability to administer the TOE locally;
  - Ability to configure the local session inactivity time before session termination or locking;
  - Ability to configure the authentication failure parameters for FIA_AFL.1;

].

5.2.4.6 FMT_SMR.2 RESTRICTIONS ON SECURITY ROLES

FMT_SMR.2.1
The TSF shall maintain the roles:

- *Security Administrator.*

FMT_SMR.2.2
The TSF shall be able to associate users with roles.

FMT_SMR.2.3
The TSF shall ensure that the conditions

- *The Security Administrator role shall be able to administer the TOE remotely* are satisfied.

5.2.5 PROTECTION OF THE TSF (FPT)

5.2.5.1 FTP_APW_EXT.1 PROTECTION OF ADMINISTRATOR PASSWORDS

FPT_APW_EXT.1.1
The TSF shall store administrative passwords in non-plaintext form.

FPT_APW_EXT.1.2
The TSF shall prevent the reading of plaintext administrative passwords.

5.2.5.2 FPT_SKP_EXT.1 PROTECTION OF TSF DATA (FOR READING OF ALL SYMMETRIC KEYS)

FPT_SKP_EXT.1.1
The TSF shall prevent reading of all pre-shared keys, symmetric keys, and private keys.

5.2.5.3 FPT_STM_EXT.1 RELIABLE TIME STAMPS

FPT_STM_EXT.1.1
The TSF shall be able to provide reliable time stamps for its own use.

FPT_STM_EXT.1.2
The TSF shall [allow the Security Administrator to set the time].

5.2.5.4 FPT_TST_EXT.1 TSF TESTING

FPT_TST_EXT.1.1 [TD0836]
The TSF shall run a suite of the following self-tests:

- *During initial start-up (on power on) to verify the integrity of the TOE firmware and software;*
- *Prior to providing any cryptographic service and* [at no other time] *to verify correct operation of cryptographic implementation necessary to fulfil the TSF;*
- [no other] *self-tests* [*none*].

to demonstrate the correct operation of the TSF.

FPT_TST_EXT.1.2
The TSF shall respond to [all failures] by [*[preventing bootup]*].

5.2.5.5 FPT_TUD_EXT.1 TRUSTED UPDATE

FPT_TUD_EXT.1.1
The TSF shall provide *Security Administrators* the ability to query the currently executing version of the TOE firmware/software and [the most recently installed version of the TOE firmware/software].

FPT_TUD_EXT.1.2
The TSF shall provide *Security Administrators* the ability to manually initiate updates to TOE firmware/software and [no other update mechanism].

FPT_TUD_EXT.1.3
The TSF shall provide means to authenticate firmware/software updates to the TOE using a [digital signature] prior to installing those updates.

5.2.6 TOE ACCESS (FTA)

5.2.6.1 FTA_SSL_EXT.1 TSF-INITIATED SESSION LOCKING

FTA_SSL_EXT.1.1
The TSF shall, for local interactive sessions, [terminate the session] after a Security Administrator-specified time period of inactivity

5.2.6.2 FTA_SSL.3 TSF-INITIATED TERMINATION

FTA_SSL.3.1
The TSF shall terminate **a remote** interactive session after a *Security Administrator-configurable time interval of session inactivity*.

5.2.6.3 FTA_SSL.4 USER-INITIATED TERMINATION

FTA_SSL.4.1
The TSF shall allow ~~user~~ **Administrator**-initiated termination of the ~~user's~~ **Administrator's** own interactive session.

5.2.6.4 FTA_TAB.1 DEFAULT TOE ACCESS BANNERS

FTA_TAB.1.1
Before establishing ~~a~~ **an administrative** user session the TSF shall display **a Security Administrator-specified** advisory **notice and consent** warning message regarding ~~unauthorised~~ use of the TOE.

5.2.7 TRUSTED PATH/CHANNELS (FTP)

5.2.7.1 FTP_ITC.1 INTER-TSF TRUSTED CHANNEL

FTP_ITC.1.1
The TSF shall **be capable of using [*TLS*] to** provide a **trusted** communication channel between itself and ~~another trusted IT product~~ **authorized IT entities supporting the following capabilities: audit server, [*[video switch control system (such as Evertz MAGNUM)]*]** that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from ~~modification or~~ disclosure **and detection of modification of the channel data**.

FTP_ITC.1.2
The TSF shall permit [the TSF, the authorized IT entities] to initiate communication via the trusted channel.

FTP_ITC.1.3
The TSF shall initiate communication via the trusted channel for *[auditing services].*

5.2.7.2 FTP_TRP.1/ADMIN TRUSTED PATH

FTP_TRP.1.1/Admin
The TSF shall **be capable of using [TLS, HTTPS] to** provide a communication path between itself and **authorized** remote **Administrators** ~~users~~ that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from disclosure **and provides detection of modification of the channel data**.

FTP_TRP.1.2/Admin
The TSF shall permit remote **Administrators** ~~users~~ to initiate communication via the trusted path.

FTP_TRP.1.3/Admin
The TSF shall require the use of the trusted path for *initial Administrator authentication and all remote administration actions*.

5.3 TOE SFR DEPENDENCIES RATIONALE FOR SFRS

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

5.4 SECURITY ASSURANCE REQUIREMENTS

The TOE assurance requirements for this ST are taken directly from the PP, which is derived from Common Criteria Version 3.1, Revision 5. The assurance requirements are summarized in Table 13.

Table 13 – Security Assurance Requirements

Assurance Class	Assurance Components	Component Description
Security Target	ASE_CCL.1	Conformance claims
	ASE_ECD.1	Extended components definition
	ASE_INT.1	ST introduction
	ASE_OBJ.1	Security objectives for the operational environment
	ASE_REQ.1	Stated security requirements
	ASE_SPD.1	Security problem definition
	ASE_TSS.1	TOE summary specification
Development	ADV_FSP.1	Basic functional specification
Guidance Documents	AGD_OPE.1	Operational user guidance
	AGD_PRE.1	Preparative procedures
Life Cycle Support	ALC_CMC.1	Labelling of the TOE
	ALC_CMS.1	TOE CM coverage
Tests	ATE_IND.1	Independent testing – conformance

Assurance Class	Assurance Components	Component Description
Vulnerability Assessment	AVA_VAN.1	Vulnerability survey

5.5 ASSURANCE MEASURES

The TOE satisfied the identified assurance requirements. This section identifies the Assurance Measures applied by Evertz's Microsystem Ltd. to satisfy the assurance requirements. The following table lists the details.

Table 14 – TOE Security Assurance Measures

SAR Component	How the SAR will be met
ASE_TSS.1.1C Refinement	The TOE summary specification shall describe how the TOE meets each SFR. In the case of entropy analysis, the TSS is used in conjunction with required supplementary information on Entropy.
ADV_FSP.1	The functional specification describes the external interfaces of the TOE; such as the means for a user to invoke a service and the corresponding response of those services. The description includes the interface(s) that enforces a security functional requirement, the interface(s) that supports the enforcement of a security functional requirement, and the interface(s) that does not enforce any security functional requirements. The interfaces are described in terms of their purpose (general goal of the interface), method of use (how the interface is to be used), parameters (explicit inputs to and outputs from an interface that control the behavior of that interface), parameter descriptions (tells what the parameter is in some meaningful way), and error messages (identifies the condition that generated it, what the message is, and the meaning of any error codes).
AGD_OPE.1	The Administrative Guide provides the descriptions of the processes and procedures of how the administrative users of the TOE can securely administer the TOE using the interfaces that provide the features and functions detailed in the guidance.
AGD_PRE.1	The Installation Guide describes the installation, generation, and startup procedures so that the users of the TOE can put the components of the TOE in the evaluated configuration.
ALC_CMC.1 ALC_CMS.1	The Configuration Management (CM) documents describe how the consumer identifies the evaluated TOE. The CM documents identify the configuration items, how those configuration items are uniquely identified, and the adequacy of the procedures that are used to control and track changes that are made to the TOE. This includes details on what changes are tracked and how potential changes are incorporated.
ATE_IND.1	Vendor will provide the TOE for testing.
AVA_VAN.1	Vendor will provide the TOE for testing. Vendor will provide a document identifying the list of software and hardware components.

6. TOE SUMMARY SPECIFICATION

6.1 TOE SUMMARY SPECIFICATION SFR DESCRIPTION

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

6.1.1 SECURITY AUDIT (FAU)

6.1.1.1 FAU_GEN.1 AUDIT DATA GENERATION AND FAU_GEN.2 USER IDENTITY ASSOCIATION

TOE generates audit logs that consist of various auditable events or actions. This includes logins, use of trusted channel/path and cryptographic operations. Each audit event contains an associated date/time stamp, a label for the type of event, a user ID (if applicable), a description of the event and any additional information specified in column three of Table 10.
Audit records are created when an auditable event that belongs to a set of predefined events has occurred. The set of auditable events can be sub-categorized into functional events and access events. The TSF generates audit records for the following events:

- Startup and shutdown of the audit function
- Administrative login and logout events
- Changes to TSF data related to configuration changes
- Generation of a CSR and associated keypair
- Addition, removal or replacement of a CA certificate
- Resetting passwords
- Failure to establish a HTTPS/TLS session
- Failure to establish a TLS session
- All use of the identification and authentication mechanism (local and remote connections to the TSF)
- Unsuccessful attempts to validate a certificate
- Initiation of a software update
- Result of a software update
- Changes to the time
- Modification of the behavior of the TSF
- Initiation and termination of the trusted channel
- Initiation and termination of the trusted path
- Attempts to unlock an interactive session
- Termination of a session by the session locking mechanism
- Configuration of local audit settings

Each audit record includes the date and time, type, subject identity (IP address, hostname, and/or username), the outcome (success or failure), and any additional information specified in column three of Table 10. The TOE only stores one certificate chain to support TLS. No other server certificates are stored. In the logs of Administrator actions which involves cryptographic keys (generating or deleting keys), the audit log will refer to the key as the "server private key".

When a user generates a CSR via WEBGUI, a new key is created without affecting the existing one. The current key remains in use until a matching new certificate is uploaded. The new key, certificate, and chain only replace the existing ones after a reboot.

6.1.1.2 FAU_STG_EXT.1 PROTECTED AUDIT EVENT STORAGE (CPP_ND_V3.0E)

The TOE is a standalone TOE. The TOE stores audit logs internally. The internal logs are stored unencrypted, but they are only accessible (export) via the web browser and view locally via console, which can only be used by Administrators. Logs are written to messages (log) file on /var/log and synced to /ssd/syslog/current live with a delay of 11 seconds. Also during every reboot the /ssd/syslog/current is renamed to /ssd/syslog/<SHUTDOWN_DATE_TIME> during shutdown and a new /ssd/syslog/current is created during startup.

The TOE does not allow any deletion or modification of the logs via the web browser or local console.

The size limit for /var/log/ folder depends on the size of the memory used on each model. This folder can also contain files other than messages (syslog files), hence, the amount of audit logs that can be saved in the /var/log/ directory can vary. The current audit log is saved in the file name 'messages'.

Within the TOE, the /var/log directory is maintained in volatile memory, whereas the /ssd/syslog directory resides on non-volatile disk storage. The content of the messages file in /var/log/ is synchronized to the /ssd/syslog/current/messages file every 11 seconds . This file is periodically compressed and archived to the disk-based storage location. Once the current messages file reaches 60MB, it will be saved as messages.0 and a new messages file will be generated to capture the new audit logs. The disk storage is configured to accommodate up to 101 compressed log files, ranging from messages.0.gz to messages.99.gz, or a cumulative total of 880 MB whichever limit is reached first.

The TOE overwrites previous audit records on a circular (FIFO) basis when both the volatile /var/log and non-volatile /ssd/syslog/current storage space for audit are full. Upon TOE's reboot, power-off, or power failure, the local audit logs remain persistent.

Logs information is also sent to an external Syslog server via 'Syslog over TLS using TLS v1.2'. Logs are sent to the Syslog servers in real-time. For this to happen, an external syslog server should be configured (IP address/TCP Port number). A trusted certificate chain that is used to sign syslog server's certificate must also be uploaded to TOE. The [EXE CC Admin Guide] explains how to configure this connection. The TOE temporarily queues a limited number of syslog messages in memory when the designated syslog destination server is unreachable. Once connectivity to the destination server is restored, the TOE transmits the queued messages from memory to ensure log continuity. Syslog messages are stored to memory(/var/log) and periodically compressed and backed up to disk(/ssd/syslog/current) along with streaming syslog in real-time.

6.1.2 CRYPTOGRAPHIC SUPPORT (FCS)

6.1.2.1 FCS_CKM.1 CRYPTOGRAPHIC KEY GENERATION

The TSF supports generation of 2048-bit RSA keys for digital signatures in support of TLS sessions (FCS_TLSC_EXT.1 and FCS_TLSS_EXT.2) and the server certificate (FIA_X509_EXT.3).

Generation of ECDSA keys with NIST curves of P-256 or P-384 or P-521 are also used to generate EC DH components for key establishment in TLS sessions (FCS_TLSC_EXT.1 and FCS_TLSS_EXT.2).

The relevant NIST CAVP certificate numbers are listed in Table 15.

6.1.2.2 FCS_CKM.2 CRYPTOGRAPHIC KEY ESTABLISHMENT

The TOE acts as both sender and recipient for elliptic curve Diffie-Hellman and RSA key establishment schemes that respectively meet the following:

- NIST Special Publication (SP) 800-56A revision 3, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" – for FCS_TLSC_EXT.1 connections to the audit server, FCS_TLSS_EXT.1 connections to the remote administrators managing the TOE over web-GUI and FCS_TLSS_EXT.2 connections to the MAGNUM server.

and

- RSA-based key establishment schemes that meet the following: RSAES-PKCS1-v1_5 as specified in Section 7.2 of RFC 8017, "Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.2. The TOE uses RSA-based key establishment for backwards compatibility for FCS_TLSC_EXT.1 connections to audit server and FCS_TLSS_EXT.2 connections to the MAGNUM server

In the case of a decryption error, the TOE response is dependent on the stage of the connection process. If the connection has not been established, the TOE prevents a connection from occurring. If the connection has already been established, the TOE drops the packet(s) in question and logs the error internally.
To address the issue of side-channel attacks, the TOE does not reveal the particular error that occurred through other channels, either through message content or timing variations.

The relevant NIST CAVP certificate numbers are listed in Table 15.

6.1.2.3 FCS_CKM.4 CRYPTOGRAPHIC KEY DESTRUCTION

Cryptographic keys are destroyed by first overwriting the key file content with zeros. A read-verification is then performed to ensure that the entire content has really been changed to zeros and not any other values. If these steps fail, then the file will be overwritten again with zeros until the read-verify step succeeds.

A sudden, unexpected power outage could disrupt zeroization and cause keys to not be zeroized. There are no other known circumstances where the TOE would not conform to these requirements.

The keys/CSPs used by the TOE, their storage location and format, and their associated zeroization method are listed in the Table 16– Key Storage and Zeroization

To delete the plain-text keys stored on the non-volatile NOR flash storage listed in the Table 16, direct interface/access via CLI is provided to view or modify the contents of these files. The CLI provides Security Administrators with a menu item to destroy all CSPs, which would initiate key destruction.

No direct interface/access is provided to view or modify the contents of the keys stored in the volatile memory.

The TOE stores private keys in the file system in non-plaintext form protected by disk encryption. In the evaluated configuration, the 'recovery' user is limited to a menu-driven interface with no shell access, ensuring that no user or administrator can access files on or run process tracing commands to inspect process memory. Keys are cleared by overwriting the contents with zeros when no longer needed.

The TLS session keys stored on RAM are automatically destroyed when the TLS session ends.

The DRBG state is zeroized using a single overwrite of zeros when the TSF is shutdown or restarted.

The above destruction methods are followed in all configurations and circumstances.

6.1.2.4 FCS_COP.1/DATAENCRYPTION CRYPTOGRAPHIC OPERATIONS (AES DATA ENCRYPTION/DECRYPTION)

The TOE provides AES encryption/decryption in CBC or GCM mode with 128- and 256-bit keys. The TOE provides AES encryption and decryption in support of TLS v1.2. The TSF performs AES encryption and decryption in CTR mode with 256-bit keys for Random Bit Generation.

The relevant NIST CAVP certificate numbers are listed in Table 15.

6.1.2.5 FCS_COP.1/SIGGEN CRYPTOGRAPHIC OPERATION (SIGNATURE GENERATION AND VERIFICATION)

The TOE supports signature generation and verification with RSA (2048-bits) with SHA-1/256/384 in accordance with FIPS PUB 186-4.

These signatures support TLS authentication and firmware verification. The TOE's server certificate is 2048-bits.

The relevant NIST CAVP certificate numbers are listed in Table 15 .

6.1.2.6 FCS_COP.1/HASH CRYPTOGRAPHIC OPERATIONS (HASH ALGORITHM)

The TOE implements hashing in byte-oriented mode. The TOE provides cryptographic hashing services in support of TLS for SHA-1, SHA-256, SHA-384 and SHA-512. SHA-256 is used in firmware integrity checks during power-on-self-tests and upgrade where the hashed-value of the images is signed with Evertz's private key and the result file (signature) is included in the firmware package file. During upgrade, the signature file is first decrypted using the public key stored on TOE, then the hashed value is re-calculated from the uploaded image file and then compared with the decrypted hash value. These hashes must match for this validation to succeed.

The locally stored passwords are salted using SHA-256. Key generation using CTR_DRBG is performed using SHA-256 as specified in NIST SP 800-90 DRBG.

The relevant NIST CAVP certificate numbers are listed in Table 15.

6.1.2.7 FCS_COP.1/KEYEDHASH CRYPTOGRAPHIC OPERATION (KEYED HASH ALGORITHM)

Keyed-hash message authentication is used as part of TLS protocol as part of the negotiated cipher suites between peers.
It is also used for firmware image integrity check where the hashed-value (HMAC-SHA-256) of the images is signed with Evertz's private key and the result file (signature) is included in the firmware package file. During upgrade, the signature file is first decrypted using the public key stored on EXE, then the hashed value (HMAC-SHA-256) is re-calculated from the uploaded image file and then compared with the decrypted hash value (HMAC-SHA-256). These hashes must match for this validation to succeed.

The following keyed-hash message authentication are used by TOE:

- HMAC-SHA-1 with 160-bit key, message digest size of 160 bits and 160 bits message block size,
- HMAC-SHA-256 with 256-bit keys, message digest sizes of 256 bits, and block size of 512 bits, and
- HMAC-SHA-384 with 384-bit keys, message digest sizes of 384 bits, and block size of 1024 bits

The relevant NIST CAVP certificate numbers are listed in Table 15.

6.1.2.8 FCS_HTTPS_EXT.1 HTTPS PROTOCOL

The TOE acts as a TLS/HTTPS server to provide web access to administrators. The TOE's HTTPS functionality is in accordance with all should statements in RFC 2818. The TOE is always the receiver of HTTPS connections. The TOE's HTTPS protocol complies with RFC 2818 by implementing all the "MUST", "REQUIRED", and "SHOULD" statements. The TOE does not implement any "MUST NOT" or "SHOULD NOT" statements. The TOE initiates an exchange of closure alerts before closing a connection. The TOE does not implement an "incomplete close".

6.1.2.9 FCS_RBG_EXT.1 RANDOM BIT GENERATION

The TOE implements a DRBG in accordance with ISO/IEC 18031:2011 using a CTR_DRBG with AES. The TSF seed the CTR_DRBG using 384-bits of data that contains at least 256 bits of entropy. The TSF gathers and pools entropy from one software-based noise source: jitter entropy (Jent v3.6.1). The entropy sources are discussed in greater detail in the Entropy Assessment documentation.

The relevant NIST CAVP certificate numbers are listed in Table 15.

6.1.2.10 FCS_TLSC_EXT.1 TLS CLIENT PROTOCOL

The TOE acts as a client when connecting to the syslog server via Syslog over TLS. The TLS server port is configurable to 6514. The TOE supports TLS communications with authentication using X.509v3 certificates.

The TSF only supports TLSv1.2 for TLS. Connection requests that include SSL 2.0, SSL 3.0, TLS 1.0 or TLS 1.1 are denied. If the TSF receives a ClientHello message that requests TLSv1.1 or earlier, the TSF sends a fatal handshake_failure message and terminates the connection.

For all the TLS client connections, if the certificate verification fails for any other reason (including a failure to establish a connection), the connection attempt fails, and the trusted channel is not established. There are no fallback authentication functions for failed certificate validation. TLS v1.2 is used to export audit records to syslog server.

The TOE specifies only a restricted set of cipher suites that it supports during the negotiation phase with a server. If no match of cipher suites can be found with peer, TLS session will not be established. These ciphersuites cannot be configured or changed by an Administrator.

The following cipher suites are supported for Syslog Server:

- TLS_RSA_WITH_AES_128_CBC_SHA
- TLS_RSA_WITH_AES_256_CBC_SHA
- TLS_RSA_WITH_AES_128_CBC_SHA256
- TLS_RSA_WITH_AES_256_CBC_SHA256
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384

TOE supports cipher suites that use ECDHE and RSA schemes for key exchange and RSA keys for authentication. These keys are generated with OpenSSL's RSA command line internally to the TSF. Elliptic curve Diffie Hellman and RSA are supported for key establishment in TLS for client. The RSA key establishment and authentication uses 2048 bits. EC-DH key establishment uses NIST curves, P-256, P-384, and P-521. By default, the TOE presents the supported Elliptic Curve Extensions, secp256r1, secp384r1, and secp521r1 in the Client Hello. The TOE conforms to RFC 5246, section 7.4.3 for key exchange. All ciphers and elliptic curves on the TOE are hardcoded. The TOE does not accept any EC Diffie-Hellman (ECDHE) parameters other than those explicitly supported.

When validating the server's certificate in authentication using X.509v3 certificates, TOE uses CRL (certification revocation list) to check for invalid certificates. The TOE fetches the CRL from the CRL Distribution Point (CRL-DP) and uses it to verify the revocation status of peer certificates during the certificate validation process.

TOE allows configuration of an RFC 6125 reference identifier from a peer it expects to connect with before connection is made.

The reference identifier is matched to the SAN in the certificate presented for authentication.

The verification against peer certificate is implemented within OpenSSL using a bitwise comparison of the SAN-DNS field. IP addresses are not supported as reference identifiers. TOE supports FQDN identifier types only. SRV-ID and URI-ID types are not supported.

The TOE does mandate SAN extension and checks SAN extension over CN when present. The TOE ignores CN when SAN is present. When SAN is not present, the TOE does not check for CN. If the SAN is incorrect, the certificate will be rejected. It will not check for the CN.

The TOE supports wildcard in certificates. The wildcard must be in the left-most label of the presented identifier. And the wildcard only covers one level of subdomains. For the reference identifier without a left-most label as in the certificate, the connection will fail, i.e., awesome.com doesn't match *.awesome.com.
The TOE does not support certificate pinning.

TOE does not support session resumption or session tickets.

The TOE does not support TLS 1.3 and therefore out-of-band provisioning of pre-shared keys (PSKs) is not supported.

The TOE supports the following signature_algorithms extension for TLS connections:

- rsa_pkcs1 with sha256(0x0401),
- rsa_pkcs1with sha384(0x0501),
- rsa_pkcs1 with sha512(0x0601),
- rsa_pss_rsae with sha256(0x0804),
- rsa_pss_rsae with sha384(0x0805),
- rsa_pss_rsae with sha512(0x0806),

The TOE, acting as a TLS client, includes the signature_algorithms extension in its Client Hello to specify supported cryptographic algorithms. By default, this extension is enabled and populated with the algorithms claimed in the ST.

6.1.2.11 FCS_TLSS_EXT.1 TLS SERVER PROTOCOL AND FCS_TLSS_EXT.2 TLS SERVER SUPPORT FOR MUTUAL AUTHENTICATION

The TOE acts as a TLS/HTTPS server to provide web access to administrators. The TOE's HTTPS functionality is in accordance with all should statements in RFC 2818. The TLS/HTTPS server accepts the connection over the TCP port 443. The TOE supports TLS communications without mutual authentication using X.509v3 certificates.

The TOE also acts as a TLS server when connecting to a video switch control system. For video switch control systems TLS trusted channels, the TOE requires TLS with mutual authentication using X.509v3 certificates. The TLS Server with mutual authentication connects over the TCP port 9672.

The TSF only supports TLSv1.2 for HTTPS/TLS. Connection requests that include SSL 2.0, SSL 3.0, TLS 1.0 or TLS 1.1 are denied. If the TSF receives a Client Hello message that requests TLSv1.1 or earlier, the TSF sends a fatal handshake_failure message and terminates the connection.

The TOE acts as the server in connections with remote administrators and with the MAGNUM video switch control system. For TLS server connections when connecting to a MAGNUM video switch control system, if a client does not present a certificate, or if the presented certificate fails verification for any other reason (including a failure to establish a connection), the connection attempt fails, and the trusted channel is not established. There are no fallback authentication functions for failed certificate authentication. TLS v1.2 is used to communicate with a MAGNUM video switch control system and support remote administrator connections.

The TOE specifies only a restricted set of cipher suites that it supports during the negotiation phase with a client. If no match of cipher suites can be found with peer, TLS session will not be started. These ciphersuites cannot be configured or changed by an Administrator.

The following cipher suites are supported for Web GUI:

- TLS_RSA_WITH_AES_128_CBC_SHA
- TLS_RSA_WITH_AES_256_CBC_SHA
- TLS_RSA_WITH_AES_128_CBC_SHA256
- TLS_RSA_WITH_AES_256_CBC_SHA256
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384

The following cipher suites are supported for MAGNUM Server :

- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384

TOE supports cipher suites that use ECDHE and RSA schemes for key exchange and RSA keys for authentication. These keys are generated with OpenSSL's RSA command line internally to the TSF. The RSA key establishment and authentication use 2048 bits. EC-DH key establishment uses NIST curves, P-256, P-384, and P-521. By default, the TOE presents the supported Elliptic Curve Extensions, secp256r1, secp384r1, and secp521r1 in the Client Hello. The TOE conforms to RFC 5246, section 7.4.3 for key exchange. All ciphers and elliptic curves on the TOE are hardcoded. The TOE does not accept any EC Diffie-Hellman (ECDHE) parameters other than those explicitly supported.

When validating a client's certificate in mutual authentication using X.509v3 certificates, TOE uses CRL (certification revocation list) to check for invalid certificates. The TOE pulls the CRL file from the CRL-DP to use by TOE during certificate validation process to check for revocation status of the peer certificates.

TOE allows configuration of an RFC 6125 reference identifier from a peer it expects to connect with before connection is made.

The reference identifier is matched to the SAN in the certificate presented for authentication.

The verification against peer certificate is implemented within OpenSSL using a bitwise comparison of the SAN-DNS field. IP addresses are not supported as reference identifiers. TOE supports FQDN identifier types only. SRV-ID and URI-ID types are not supported.

The TOE does mandate SAN extension and checks SAN extension over CN when present. The TOE ignores CN when SAN is present. When SAN is not present, the TOE does not check for CN. If the SAN is incorrect, the certificate will be rejected. It will not check for the CN.

TOE does not support session resumption.

The TOE does not support TLS 1.3 and therefore out-of-band provisioning of pre-shared keys (PSKs) is not supported.

The TOE supports the following signature_algorithms extension for TLS connections:

- rsa_pkcs1 with sha256(0x0401),
- rsa_pkcs1with sha384(0x0501),
- rsa_pkcs1 with sha512(0x0601),
- rsa_pss_rsae with sha256(0x0804),
- rsa_pss_rsae with sha384(0x0805),
- rsa_pss_rsae with sha512(0x0806),

If the claimed signature_algorithm is present in the Client Hello, then the connection is accepted otherwise denies the connection. By default, the signature_algorithms extension is supported on the TOE.

6.1.3 IDENTIFICATION AND AUTHENTICATION (FIA)

6.1.3.1 FIA_AFL.1 AUTHENTICATION FAILURE HANDLING

An administrator can configure the number of unsuccessful attempts a remote administrator can make before a lock-out occurs and can enable/disable the length of time that the remote administrator is locked out. The attempts can range between 3 and 20. The default number of attempts is 10. The length of time cannot be configured, with a default of 12 hours.

If a user enters an incorrect password the configured number of times, the user is locked out, and they cannot log in through any remote interface on the TOE. The username will show **"**Lockout Enabled" on the Settings → Users page on the web interface.

The TSF also implements an increasing wait time for each unsuccessful login attempt. It is not configurable.

Users must have an administrator unlock their account before they can regain access. Administrators can also have a different administrator unlock their account.

Lockouts are not enforced on the TOE's console interface. This ensures that authentication failures cannot lead to a situation where no administrator access is available.

6.1.3.2 FIA_PMG_EXT.1 PASSWORD MANAGEMENT

TOE enforces that passwords must meet minimum requirements such as length, mix of number, lower/upper case letters, and the following special characters *"!"; "@"; "#"; "$"; "%"; "^"; "&"; "*"; "("; ")";"~"; "`"; "_"; "-"; "+"; "="; "{"; "["; "}"; "]"; "|"; "\"; ":"; ";"; ["]; ['];* *"<"; ","; ">"; "."; "?"; "/"; [space]*.
At least two characters from each category are required (upper case letter, lower case letter, number special character). Administrators can configure a minimum password length between 15 and 20 characters.

6.1.3.3 FIA_UIA_EXT.1 USER IDENTIFICATION AND AUTHENTICATION

The accounts that the TOE will establish are Security Administrator accounts. Users only control the TOE indirectly via MAGNUM. Administrators are identified and authenticated via username and password prior to performing any operations other than acknowledging the warning banner. The TOE Administrators user accounts module maintains Security Administrator credentials. Security administrators can have different management roles assigned to create restricted users.

Administrators are able to login via the web interface using HTTPS or locally on the serial port. A username and a password are required to authenticate the administrator for both methods. The Security Administrator is considered authenticated if the username and password match the stored credential values. For serial console, only the administrator has access to view logs under /var/log and /ssd/current path.

Prior to successful identification and authentication on all interfaces (Web GUI and Local Console), the TSF displays the TOE access banner specified in FTA_TAB.1. Users must acknowledge the warning banner before they can login to the system.

6.1.3.4 FIA_UAU.7 PROTECTED AUTHENTICATION FEEDBACK (REFINEMENT)

When the user is entering their password over the local console, the TSF does not echo any characters back.

6.1.3.5 FIA_X509_EXT.1/REV X.509 CERTIFICATE VALIDATION

TOE uses OpenSSL for X.509 certificate validation. The certificate path is validated by ensuring that all the CA certificates have the basicConstraints extension, validity, issuer, authority_key identifier and the path must terminate with a trusted CA certificate. The extendedKeyUsage on each certificate is also checked to ensure there is no inappropriate usage. Server certificates must have the Server Authentication purpose, client's certificates must have the Client Authentication purpose. Certificates for code signing and OCSP signing are not used or accepted by the TOE. Each certificate (other than the leaf certificate) in the certificate chain has the basicConstraints extension with the CA flag set to TRUE (Subject Type = CA). Certificates are not used for any purposes other than establishing TLS sessions.

When the TOE acts as a server for the WebGUI, it does not perform verification of its server certificate. It performs verification of every certificate in the chain upon upload. When the TOE acts as a server for the MAGNUM connection, it performs verification of every certificate in the chain for client authentication as well as upon upload. The certificate presented by remote TLS clients using mutual authentication is validated during the

establishment of a TLS connection. The full certificate chain presented by TLS servers are validated during the establishment of a TLS connection.

For an expired certificate, TOE will deny the connection. TOE also uses CRL to verify whether the leaf certificate or intermediate CA certificate have been revoked. During session establishment with TOE, any byte modification in the certificate will lead to the failure of connection.
The TSF verifies the validity of a certificate when:

- A TLS client establishes a TLS connection with mutual authentication.
- A TLS server presents certificates to the TOE as a part of a TLS connection.

If the Security Administrator loads a certificate that includes the basicConstraints extension with the CA flag set to TRUE (Subject Type = CA), the TSF does not perform certificate path validation.

6.1.3.6 FIA_X509_EXT.2 X.509 CERTIFICATE AUTHENTICATION

Instructions about generating/downloading CSR and loading certificate can be found on [EXE CC Admin Guide]. The Administrator has the ability to upload either a complete certificate chain or an individual CA certificate to the EXE trust store.
The same certificate will be used by TOE for both web service and MAGNUM control. The same CA will be used for certificate verification. TOE enforces mutual authentication and therefore requires client certificates to establish a connection.

The CRLs are obtained from a CRL distribution point over HTTP and are refreshed on every connection establishment. If the TOE is unable to reach the CRL DP it will reject the certificate and the session associated with the certificate will be rejected, a log is generated indicating the reason for validation failure.

6.1.3.7 FIA_X509_EXT.3 X.509 CERTIFICATE REQUESTS

The TSF allows Security Administrators to generate Certificate Signing Requests.
The TSF requires the Security Administrator to specify the following values:

- Common Name
- Organization
- State or Province
- Organization Unit
- Locality
- Country
- Email Address
- Subject Alt Name

A CSR can be generated only via the webGUI. When validating certificates, each certificate from the chain is sequentially validated, terminating at the root CA. If any invalid certificate is found in this process, the validation fails.

6.1.4 SECURITY MANAGEMENT (FMT)

6.1.4.1 FMT_MOF.1/FUNCTIONS MANAGEMENT OF SECURITY FUNCTIONS BEHAVIOUR

The TOE restricts the ability to enable and disable the transmission of audit records to an external audit server to Security Administrators. The WebEasy interface allow the Security Administrator to perform the following TSF management functions:

- View/Edit settings for sending audit data to the Syslog Server

Administrator can modify the behavior of transmitting audit data to an external audit server with the following configuration from the WebEasy interface:

- Login to the TOE Management Web Application.
- Click "General" menu on top of the displayed page.
- Click on "Info/Logging" Tab at bottom of right-hand side of the page.
- Under "Sys Log 1" of "Log Streaming" section, enter the following information:
  - Destination IP Address
  - Destination Port
  - Level
  - Reference Identifier

6.1.4.2 FMT_MOF.1/MANUALUPDATE MANAGEMENT OF SECURITY FUNCTIONS BEHAVIOR

There are no specific requirements for non-distributed TOEs.

6.1.4.3 FMT_MTD.1/COREDATA MANAGEMENT OF TSF DATA

The TOE implements Role Based Access Control (RBAC). Security Administrative must login before they can access any administrative functions.

The TOE maintains the following roles:

- Administrator
- rw-user
- ro-user

The Security Administrator is the only account that should be used with the role of this account set to administrator role. There are no limitations/restrictions for the administrator role. There is no option to edit or modify the administrator role access. The Administrator has the ability to grant or remove permissions (which are also called 'Restrictions') to both ro-users and rw-users. By default, users with rw-user role can change the configuration of TOE, view the event log, and perform firmware upgrades. However, they are not authorized to change general system settings, manage user accounts, modify roles, or create new users. All permissions granted to rw-user role are subject to further restriction by the TOE's Security Administrator. As part of the TOE's evaluated configuration, it is required that the "upgrade" permission be disabled for the rw-user role. The users with ro-user role cannot change any TOE configuration settings, user settings or perform firmware upgrades. This role can only view TOE and user settings and view the event log.

No administrative functionality is available prior to login. The TSF displays a warning banner prior to user authentication. Information on how a Security Administrator can manage Audit Operations is described in the TSS section of FAU_STG_EXT.1.

The TOE maintains a trust store that contains the TOE's CA certificate. Access to the trust store is restricted to Security Administrators through role-based access controls. Only Security Administrators can upload a certificates chain, and uploading a new chain replaces the previously installed certificate chain.

6.1.4.4 FMT_MTD.1/CRYPTOKEYS MANAGEMENT OF TSF DATA

The WEB GUI allow the Security Administrator to perform the following TSF management functions on cryptographic keys:

- TLS Key Generation (TLS keys are automatically generated when creating a CSR)
- TLS Key Reset/Replacement (when a CSR is generated, previous RSA 2048-bit key will be deleted and replaced by the new key. RSA 2048-bit key cannot be imported from outside the TOE. The administrators cannot delete TLS keys manually).

The administrative interfaces provided by the TSF do not allow any of these functions to be accessed by unauthenticated or unauthorized users.

6.1.4.5 FMT_SMF.1 SPECIFICATION OF MANAGEMENT FUNCTIONS

TOE gives the Security Administrator the ability to manage the security functions: auditing operations, administrative user accounts, password and session policies, advisory banners, software updates, as well as cryptographic functions. TOE ensures that only secure values are accepted for security attributes. The TSF implements the Security Administrator role to authorized administrators of the TOE. The TSF allows the Security Administrators to administer the TSF via a local CLI and a remote web interface. The TSF implements role-based access control of these management functions to users that have been identified, authenticated, and authorized with the Security Administrator role.

The web UI and CLI (local and remote) allow the Security Administrator to perform the following TSF management functions:

- Ability to administer the TOE remotely;
- Ability to configure the access banner;
- Ability to configure the remote session inactivity time before session termination;
- Ability to update the TOE, and to verify the updates using digital signature capability prior to installing those updates;
- Ability to modify the behaviour of the transmission of audit data to an external IT entity;
- Ability to manage the cryptographic keys;
- Ability to re-enable an Administrator account;
- Ability to set the time which is used for time-stamps;
- Ability to manage the TOE's trust store and designate X509.v3 certificates as trust anchors;
- Ability to generate Certificate Signing Request (CSR) and process CA certificate response;
- Ability to configure the authentication failure parameters for FIA_AFL.1;
- Ability to administer the TOE locally;
- Ability to configure the local session inactivity time before session termination or locking;

The web interface allows the Security Administrator to perform the following TSF management functions:

- Edit login banner;
- Create certificate signing request CSR, download a CSR;
- Import certificates;

- Import Trusted CA certificate;
- Delete (Replace) x509 certificates in the trust store;
- Configure webGUI and console menu system timeout;
- Verify/Install Firmware Updates;
- View/Edit settings for sending audit data to the Syslog Server;
- View/Edit authentication failure parameters;
- Unlock a locked user after the login failure threshold is exceeded;

The following can only be performed from the console interfaces:

- Configure TOE date and time;
- Control port IP configuration;
- Reset certificates.
- Zeroize all Critical Security Parameters (CSP)
- Reboot

Administrators can administer TOE locally through serial port connection. A console menu can be used to perform configurations tasks such as setting IP/system time/system reboot, etc.

Administrators can administer TOE remotely through its web interface, which runs on HTTPS. The web interface supports a broader set of configuration settings that include configurations for certificate imports, syslog server, etc.

The administrative interfaces provided by the TSF do not allow any of these functions to be accessed by unauthenticated or unauthorized users.

The local audit is not user configurable.

6.1.4.6 FMT_SMR.2 RESTRICTIONS ON SECURITY ROLES

The TOE implements Role Based Access Control (RBAC). The Security Administrator is the only account that should be used. This account has the Administrator role.
The TOE maintains the following roles:

- Administrator
- rw-user
- ro-user

The Security Administrator is the only account that should be used with the role of this account set to administrator role. There are no limitations/restrictions for the administrator role. There is no option to edit or modify the administrator role access. The Administrator has the ability to grant or remove permissions (which are also called 'Restrictions') to both ro-users and rw-users. By default, users with rw-user role can change the configuration of TOE, view the event log, and perform firmware upgrades. However, they are not authorized to change general system settings, manage user accounts, modify roles, or create new users. All permissions granted to rw-user role are subject to further restriction by the TOE's Security Administrator. As part of the TOE's evaluated configuration, it is required that the "upgrade" permission be disabled for the rw-user role. The users with ro-user role cannot change any TOE configuration settings, user settings or perform firmware upgrades. This role can only view TOE and user settings and view the event log.

When a user account is created (by administrator), it must be assigned with a role that specifies the privileges the account will have. The administrator can choose to assign an existing role with pre-defined privileges (administrative role, role with read-write user privileges, and role with read-only privileges) or create a new role with customized privileges.

6.1.5 PROTECTION OF THE TSF (FPT)

6.1.5.1 FPT_APW_EXT.1 PROTECTION OF ADMINISTRATOR PASSWORDS

Passwords are the authentication data stored by the TOE. The TSF does not store plain text passwords. Passwords are salted using SHA-256 and stored in a secure location which is not accessible to users. Secure (one-way) hash functions ensure that it's computationally impossible to recover a plaintext from its hashed value. Passwords are unable to be viewed through an interface specifically designed for that purpose.

6.1.5.2 FPT_SKP_EXT.1 PROTECTION OF TSF DATA (FOR READING OF ALL SYMMETRIC KEYS)

The TOE protects all critical keys and security parameters by restricting their storage to isolated environments that completely lack external administrative or user read interfaces. Ephemeral cryptographic assets—such as EC Diffie-Hellman, TLS session encryption (AES), and authentication (HMAC) keys—exist strictly as plaintext within volatile RAM and are never committed to disk. For persistent data, the HTTPS/TLS Server/Host private keys are secured at rest using full-disk encryption, while administrative passwords are mathematically obscured inside configuration files using one-way, randomized salted SHA-256 hashes. The firmware update private key is entirely absent from the physical device, leaving only the non-confidential RSA public key on the flash disk to verify incoming image signature.

The TSF stores data in an encrypted file system that uses a hardware-specific unique ID on the disk. It is mounted during TOE bootup time. In the evaluated configuration, the 'recovery' user is limited to a menu-driven interface with no shell access, ensuring that no user or administrator can access files or run process tracing commands to inspect process memory.

When the TOE is rebooted, all keys stored in volatile memory are subject to the clearing methods present in FCS_CKM.4.

Information regarding the storage locations, usage, and method of storage of the cryptographic keys described in FCS_CKM.4 above.

6.1.5.3 FPT_STM_EXT.1 RELIABLE TIME STAMPS

The TSF provides a reliable timestamp from the hardware clock on the TOE. Timestamps found in auditable log events use the system clock on TOE. In addition to the purpose of generating audit logs, this timestamp is used for the purposes of other time-sensitive operations on the TOE including cryptographic key regeneration intervals. Administrators can, as needed, set the system time clock through serial port console menu.

Other functions which make use of timestamps include local and remote session inactivity and verification of X.509 certificate validity periods.

The new system time is also used to set the hardware clock, which is a clock that runs independently of any control program running in the CPU and even when TOE is powered off. During TOE system startup, system time is initialized to the time from the hardware clock.

6.1.5.4 FPT_TST_EXT.1 TSF TESTING

The TSF performs the following self-tests at power-on:

- firmware integrity check that compares the SHA256 checksum of the loaded firmware with a permanently stored hash value;

- The TSF enables FIPS mode on the OpenSSL library when Secure Mode is configured. Upon enabling FIPS mode the algorithm self-tests required by FIPS are performed. The OpenSSL library self-tests include:

    - HMAC (KAT_Integrity): An HMAC-based Known Answer Test is used to verify the integrity of the cryptographic module itself, confirming the module's code has not been altered.
    - HMAC (Module_Integrity): A separate HMAC-based integrity check is performed over the software module to detect unauthorized modification prior to use.
    - SHA-1 (KAT_Digest): The SHA-1 hashing algorithm is tested using a known-answer test vector; the computed digest is compared against the expected pre-computed digest to confirm correct operation.
    - SHA-2 (KAT_Digest): The SHA-2 family hashing algorithm is tested using a known-answer test vector; the computed digest is compared against the expected pre-computed digest to confirm correct operation.
    - AES-GCM (KAT_Cipher): A known plaintext is encrypted using AES-GCM with a known key, and the resulting ciphertext is compared against the expected ciphertext; the ciphertext is then decrypted and the recovered plaintext is compared against the original to confirm correct encryption and decryption operation.
    - RSA (KAT_Signature): The RSA digital signature algorithm is tested using a known-answer test vector; the computed signature is compared against the expected pre-computed result to confirm correct operation.
    - ECDSA (KAT_Signature): The ECDSA digital signature algorithm is tested using a known-answer test vector; the computed signature is compared against the expected pre-computed result to confirm correct operation.
    - TLS12_PRF (KAT_KDF): The TLS 1.2 pseudorandom function (key derivation function) is tested using a known-answer test vector to confirm correct key material derivation.
    - CTR (DRBG): The Counter-mode Deterministic Random Bit Generator is tested using a known seed value; the generated output is compared against the expected output for that seed.
    - ECDH (KAT_KA): The Elliptic Curve Diffie-Hellman key agreement algorithm is tested using a known-answer test vector to confirm correct shared secret computation.
    - RSA_Encrypt (KAT_AsymmetricCipher) : The RSA encryption operation is tested using a known-answer test vector; the computed ciphertext is compared against the expected result.
    - RSA_Decrypt (KAT_AsymmetricCipher): The RSA decryption operation is tested using a known-answer test vector; the computed plaintext is compared against the expected result.

After loading the image, a hash value of SHA256 is computed from the memory partition containing the image. This hash value is compared with a pre-stored hash value at another location on flash. The two hash values must match for the boot process to succeed.

If any of the other checks fail, the TOE will fail to boot and an error will be displayed. Administrators are instructed to contact Evertz service department for repair if the failure does not clear on reboot. These self-tests ensure the TOE software is the correct image and that cryptographic functions are performing appropriately. If failures are seen by the Administrator, they should be immediately corrected.

6.1.5.5 FPT_TUD_EXT.1 TRUSTED UPDATE

The Security administrators do not have access to install any applications on the TOE. The TOE embedded system can only be updated with the valid firmware released by Evertz. The current firmware version is displayed on both webpage and in serial console menu. The TOE supports delayed activation of updates hence the inactive versions can be manually set to active by the Security Administrator on the Web GUI. Firmware updates are done from the TOE web GUI under "upgrade". The EXE is typically deployed in a closed network without direct access to the internet. In these instances, Administrators are required to contact Evertz to receive notification of production updates directly or via email blast. Customers requiring secure delivery for site policy can request secure courier delivery of software updates. Digital delivery of new TOE firmware and updates are provided via Microsoft OneDrive, which requires secure authentication and uses secure TLS (HTTPS).

The first step of upgrading firmware image is to choose a desired image slot index. If any firmware image is pre-installed to the desired image slot, delete it. Continue using the image slot for firmware upload. After the firmware is uploaded, TOE will verify the firmware binary header with an Evertz-EXE-specific-file format header. If there is no mismatch, the new firmware code will be parsed for valid digital signatures.

Verification of the firmware's digital signature is performed using the RSA 2048-bit public key with SHA-256 stored on the TOE. If verification is unsuccessful, the firmware update file is rejected and an error is displayed. The TSF does not provide an interface for administrators to change the locally stored public key. If verification is successful, firmware-specific files are generated. Checksums of the firmware binary and firmware-specific files are generated and stored under the selected image slot when uploading the firmware binary. The generated checksums are used to verify the firmware binary in the image slot location without compromising file integrity.

Once the desired image slot upgrade with the firmware binary is completed successfully, the administrator must manually change the "Next Boot Image" value from the current boot image to the newly installed image slot. When setting the next image, the firmware binary is extracted to the location used during boot, and boot-specific files are created. Checksums of the extracted firmware files and boot-specific files are generated to verify that the extraction process has not compromised file integrity.

A SHA-256 hash value of the image is generated and signed with Evertz's RSA 2048-bit private key. The resulting signature file is included in the firmware package together with the firmware binary. During the upgrade process, the signature file is verified using the public key stored on the EXE, the hash value is recalculated from the uploaded firmware image file, and the calculated hash is compared with the verified hash value.

If the digital signature fails, the upgrade fails, and a log event is generated. If the digital signature succeeds, the upgrade proceeds and the updated firmware is installed onto the TOE.

The administrator must manually reboot for the new update to take effect.

6.1.6 TOE ACCESS (FTA)

6.1.6.1 FTA_SSL_EXT.1 TSF-INITIATED SESSION LOCKING , FTA_SSL.3 TSF-INITIATED TERMINATION AND FTA_SSL.4 USER-INITIATED TERMINATION

The TOE provides session management by enforcing a configurable inactivity timeout and defined session handling modes. Security Administrators can configure a maximum allowable period of inactivity for an administrator

session through the web interface. The configured timeout applies to both the local console and web interfaces. The maximum session timeout allowed depends on session Handling type.

If there is no user interaction with the TOE for the configured period, the session is automatically terminated, and the user must re-authenticate to regain access. When a session is terminated due to timeout, any unsaved changes are discarded.

The TOE supports three session handling modes: Relaxed, Some, and Strict. These modes control the number of concurrent rw-user and ro-user sessions and whether multiple logins from different locations are permitted. Based on the selected mode, the TOE enforces limits on simultaneous read-write access while allowing read only access as defined by the configuration. The following are the conditions for each mode:

- Relaxed
  - Any number of RW (Read/Write) and R/O (Read-Only) users may be logged in at the same time.
  - Maximum session timeout: 9999 seconds.

- Some
  - Only 1 RW user is allowed, and the user may be logged in from multiple locations.
  - Any number of R/O users are allowed simultaneously.
  - Maximum session timeout: 600 seconds.

- Strict
  - Only 1 RW user is allowed, and the user may be logged in from only one location at a time.
  - Any number of R/O users are allowed simultaneously.
  - Maximum session timeout: 600 seconds.

Administrators may also manually terminate their own sessions by selecting "Logout" in the upper-right corner of the web interface or by typing 'x' to exit the top-level menu on the console.

Note: Relaxed mode is by default. Strict mode was tested and is part of this evaluation.

6.1.6.2 FTA_TAB.1 DEFAULT TOE ACCESS BANNERS

The TSF presents the access banner prior to authentication when a user connects to the remote web interface or local console CLI described in the FIA_UIA_EXT.1 description.

The TSF enables Security Administrators to alter the warning banner by navigating to the Perpetual User License Agreement tab on the web. From here the Security Administrator can modify the "Agree" text and/or the "Disagree" text. (The "Disagree" text shows up when a user "disagrees" with the Security Banner text. The banner can provide warnings against unauthorized access to the TOE as well as any other information that the Security Administrator wishes to communicate. Users who select "Disagree" are not permitted access to the TSF.

6.1.7 TRUSTED PATH/CHANNELS (FTP)

6.1.7.1 FTP_ITC.1 INTER-TSF TRUSTED CHANNEL

The TSF communicates with the external syslog server using Syslog over TLSv1.2 with X.509 certificate-based authentication as described in the descriptions of FAU_STG_EXT.1 and FCS_TLSC above. The TSF initiates the trusted channel with an external audit server using Syslog over TLSv1.2 protocol.

The TSF communicates with a MAGNUM (Video Switch Server) through TLS v1.2 protocol with X.509 certificate-based mutual authentication as well as described in the FCS_TLSS above. The MAGNUM initiates the trusted channel with the TOE. The TOE's routing and switching of video signals is controlled by MAGNUM via remote administration over the TCP port 9672.

6.1.7.2 FTP_TRP.1/ADMIN TRUSTED PATH

The TSF provides a trusted path for remote administration using HTTPS/TLS as described in FCS_HTTPS_EXT.1 and FCS_TLSS_EXT.1 descriptions. TOE uses encryption and restricts the choices of ciphers, hashes, and key-exchange algorithms to those allowed by the NDcPP.

6.2 CAVP ALGORITHM CERTIFICATE DETAILS

Each of these cryptographic algorithms have been validated as identified in the table below.

Table 15 – CAVP Algorithm Certificate References

SFR	Algorithm Description	Implementation Name	CAVP Alg.	CAVP Cert #
FCS_CKM.1	RSA schemes using cryptographic key sizes of [**2048 bits**] that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.3 or FIPS PUB 186-5, "Digital Signature Standard (DSS)", A.1	EXE Cryptographic Module version 1.11	RSA KeyGen (FIPS186-4) (Modulo 2048)	A7973
	ECC schemes using 'NIST curves' **[P-256, P-384, P-521]** that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4, or FIPS PUB 186-5, "Digital Signature Standard (DSS)", Appendix A.2, or ISO/IEC 14888-3, "IT Security techniques - Digital signatures with appendix - Part 3: Discrete logarithm based mechanisms", Section 6.6.	EXE Cryptographic Module version 1.11	ECDSA KeyGen (FIPS186-4) (Curve P-256,P-384,P-521) ECDSA KeyVer (FIPS186-4) (Curve P-256,P-384,P-521)	A7973
FCS_CKM.2	RSA-based key establishment schemes that meet the following: RSAES-PKCS1-v1_5 as specified in Section 7.2 of RFC 8017, "Public-Key Cryptography Standards (PKCS) #1:	EXE Cryptographic Module version 1.11	No NIST CAVP, CCTL has performed all assurance/evaluation activities and documented in	This testing was performed in conjunction with FTP_TRP.1/Admin Test #1 and FTP_ITC.1 Test

SFR	Algorithm Description	Implementation Name	CAVP Alg.	CAVP Cert #
	RSA Cryptography Specifications Version 2.2"		the ETR and AAR accordingly.	#1 to demonstrate correct operation.
	Elliptic curve-based key establishment schemes that meet the following: NIST Special Publication 800-56A Revision 3, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography"	EXE Cryptographic Module version 1.11	KAS-ECC-SSC Sp800-56Ar3 (Domain Parameter Generation Methods: P-256, P-384, P-521)	A7973
FCS_COP.1/ DataEncryption	**GCM** mode and cryptographic key sizes **128 bits, 256 bits** that meet: AES as specified in ISO 18033-3 and GCM as specified in ISO 19772.	EXE Cryptographic Module version 1.11	AES-GCM (Key length 128,256)	A7973
	CBC mode and cryptographic key sizes **128 bits, 256 bits** that meet: AES as specified in ISO 18033-3, CBC as specified in ISO 10116.		AES-CBC (Key length 128,256)	
	CTR mode and cryptographic key sizes **256 bits** that meet: AES as specified in ISO 18033-3, and CTR as specified in ISO 10116.		AES-CTR (Key length 256)	
FCS_COP.1/ SigGen	RSA Digital Signature Algorithm using key sizes of **2048 bits** that meets FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 5.5, using PKCS #1 v2.1 or FIPS PUB 186-5, "Digital Signature Standard (DSS)", Section 5.4 using PKCS #1 v2.2 Signature Schemes RSASSA-PSS and/or RSASSA-PKCS1v1_5; ISO/IEC 9796-2, Digital signature scheme 2 or Digital Signature scheme 3	EXE Cryptographic Module version 1.11	RSA SigGen (FIPS186-4) (Modulo: 2048) RSA SigVer (FIPS186-4) (Modulo: 2048)	A7973
FCS_COP.1/ Hash	[**SHA-1, SHA-256, SHA-384, SHA-512**] and message digest sizes [**160, 256, 384, 512**] bits	EXE Cryptographic Module version 1.11	SHA-1 SHA2-256 SHA2-384 SHA-512	A7973
FCS_COP.1/ KeyedHash	[**HMAC-SHA-1, HMAC-SHA- 256, HMAC-SHA-384**] and cryptographic key sizes [**key size (in bits) used in**	EXE Cryptographic Module version 1.11	HMAC-SHA-1 HMAC-SHA2-256 HMAC-SHA2-384	A7973

SFR	Algorithm Description	Implementation Name	CAVP Alg.	CAVP Cert #
	HMAC] and message digest sizes [**160, 256, 384**] bits			
FCS_RBG_EXT.1	**CTR_DRBG (AES)**	EXE Cryptographic Module version 1.11	Counter DRBG (AES-256)	A7973

6.3 CRYPTOGRAPHIC KEY DESTRUCTION

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

Table 16 – Key Storage and Zeroization.

Keys/CSPs	Purpose	Origin (Generation / Import)	Storage Location	Method of Zeroization
EC Diffie-Hellman Keys	Key agreement and key establishment	Generated per TLS handshake using an approved RNG	Plaintext in RAM	Overwritten with zeroes when no longer needed.
HTTPS/TLS Server/Host Key	RSA and EC private key used in the HTTPS/TLS protocols	Generated internally or imported via secure administrative interface	Stored in File System and protected using disk Encryption	Overwritten with zeroes when no longer needed.
HTTPS/TLS session authentication key	HMAC SHA-1, -256, or -384 key used for HTTPS/TLS session authentication.	Ephemeral; derived from Master Secret during TLS handshake	Plaintext in RAM	During boot they get erased. When the client closes TLS session, the keys get erased. Keys are erased by overwriting the contents with zeros when no longer needed.
HTTPS/TLS Session Encryption Key	AES (128, 256) key used for HTTPS/TLS session encryption	Ephemeral; derived from Master Secret during TLS handshake	Plaintext in RAM	During boot they get erased. When the client closes TLS session, the keys get erased. Keys are erased by overwriting the contents with zeros when no longer needed.
Locally Stored Passwords	User Authentication	Salt generated using approved RNG; hash = SHA-	Salted using SHA-256 in configuration file	A temporary copy is created, modified, and then replaces the old

Keys/CSPs	Purpose	Origin (Generation / Import)	Storage Location	Method of Zeroization
		256(password salt)		file when no longer needed.

7. ACRONYM TABLE

Acronyms should be included as an Appendix in each document.

Table 17 – Acronyms

Acronym	Definition
AES	Advanced Encryption Standard
CC	Common Criteria
CRL	Certificate Revocation List
DTLS	Datagram Transport Layer Security
EP	Extended Package
GUI	Graphical User Interface
IP	Internet Protocol
NDcPP	Network Device Collaborative Protection Profile
NIAP	Nation Information Assurance Partnership
NTP	Network Time Protocol
OCSP	Online Certificate Status Protocol
PP	Protection Profile
RSA	Rivest, Shamir & Adleman
SFR	Security Functional Requirement
SSH	Secure Shell
ST	Security Target
TOE	Target of Evaluation
TLS	Transport Layer Security
TSS	TOE Summary Specification
CIP SLTS	(Civil Infrastructure Platform – Super Long Term Support)
EXE CC Admin Guide	MMA10G-EXE Series Security Administrative Guide Addendum for Common Criteria
RW	Read/Write
R/O	Read Only