EVERTZ MMA10G-EXE NETWORK DEVICE COLLABORATIVE PROTECTION PROFILE SECURITY TARGET V1.0A

25 April, 2018

Evertz Microsystems Ltd
5292 John Lucas Dr. Burlington, Ontario, Canada

Table of Contents

Table of Tables

DOCUMENT REVISION HISTORY

VERSION	DATE	REVISION DESCRIPTION	AUTHOR
0.1	8/31/17	Initial Draft of Sections 1, 2, 3 and 6	Acumen Security
0.2	10/11/17	Updates based on vendor comments	Acumen Security
0.3	2/22/18	Updated based on testing	Acumen Security
0.4	3/18/18	Updated for recent TDs and product information	Acumen Security
0.5	4/2/18	Updated TDs, TD0255 archived	Acumen Security
1.0	4/21/18	Updated for public release	Acumen Security
1.0a	4/25/18	Updated to consistently reference name	Acumen Security

1 Introduction

This document demonstrates the compliance of a suite of functionally similar products of Evertz Microsystems, Ltd. with the Network Device Collaborative Protection Profile, version 1.0. The devices are the MMA10G-EXE series (hereinafter referred to as "MMA10G-EXE"), which are Ethernet switches optimized for video content.

1.1 Security Target (ST) and Target of Evaluation (TOE) Reference

ST Title	**EVERTZ MMA10G-EXE NETWORK DEVICE COLLABORATIVE PROTECTION PROFILE SECURITY TARGET**
ST Version	**1.0a**
ST Issue Date	**4/25/2018**
TOE Identification – Evertz MMA10G-EXE	
Part ID	**Firmware Version**
MMA10G-EXE-16-FR	Build 18695
MMA10G-EXE-28-FR	
MMA10G-EXE-40-FR	

Table 1. ST and TOE Reference

2 Target of Evaluation (TOE) Overview

2.1 TOE Operational Environment

The TOE operational environment includes:

- Local console terminal for management
- Management workstation with browser for administrator access via the network
- Remote syslog server receiving audit records from the TOE via the network
- Certificate Authority (CA) used in support of certificate validation operations

2.2 Physical Scope of the TOE

The MMA10G-EXE switch is a 10 Gigabit (Gb) Internet Protocol (IP) switch optimized for video-over-IP traffic (compressed or uncompressed). The three models of the MMA10G-EXE included in the evaluation provide identical functionality. The only differences between them are the physical size (16 RU, 28RU and 40RU respectively) and the number of physical interfaces supported.

The MMA10G-EXE 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. Video routing networks utilize dedicated physical plant and are highly efficient, sustainable, and secure. The MMA10G-EXE provides the same capability within the context of packet-based networks using shared network infrastructure.

Traditional packet-based networks do not support the extremely high standards for signal integrity and fault tolerance required for broadcast video. Evertz's solution to this problem has been to develop a packet-based switching fabric from a video perspective, rather than rely on traditional packet-based network architecture. Since video by nature has a unidirectional flow, and also since it is normal for multiple copies of a single incoming video stream to be sent to multiple output destinations, the MMA10G-EXE exclusively uses multicast IP addressing. Unicast is not feasible for streaming video in an enterprise production environment and is not supported by the MMA10G-EXE platform.

Multicast switching can be challenging, especially for non-automated systems. Momentary delays and signal loss are common in these networks but are unacceptable in broadcast environments. To address this issue, a typical MMA10G-EXE installation will also include a standard video routing switch software platform (such as Evertz Magnum) to route data seamlessly 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, 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 physical scope of the TOE includes the MMA10G-EXE Security Administration Manual.

2.3 Logical Scope of the TOE (Overview)

The NDcPP-compliant TOE is comprised of several security features. Each of the security features identified above consists of several security functionalities, as identified below.

1. Security audit
2. Cryptographic support
3. Identification and authentication
4. Secure Management
5. Protection of the TSF
6. TOE access
7. Trusted path/channels

These features are described in more detail in the subsections below.

2.3.1 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. Very broadly, the Audit events generated by the TOE include:

- Establishment of a Trusted Path or Channel Session
- Failure to Establish a Trusted Path or Channel Session
- Termination of a Trusted Path or Channel Session
- Failure of Trusted Channel Functions
- Identification and Authentication

- Unsuccessful attempt to validate a certificate
- Any update attempts
- Result of the update attempt
- Management of TSF data
- Changes to Time

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 has no access to edit them, only to copy or delete (clear) them. Audit records are protected from unauthorized modifications and deletions. The previous audit records are overwritten when the allocated space for these records reaches the threshold on a FIFO basis.

2.3.2 Cryptographic Support

The TOE provides cryptography support for secure communications and protection of information. The cryptographic services provided by the TOE include: symmetric encryption and decryption using AES; asymmetric key generation; cryptographic key establishment using DH key establishment; digital signature using RSA; cryptographic hashing using SHA-256; random bit generation using DRBG and keyed-hash message authentication using HMAC-SHA (SHA-256). The TOE implements the secure protocols TLS/HTTPS on the server side and TLS on the client side. The algorithm certificate references are listed in the table below.

Algorithm	Description	Mode Supported	CAVP Cert. #	Module Name	HW
AES	Used for symmetric encryption/decryption FCS_TLSC_EXT.1 FCS_TLSS_EXT.2 FCS_HTTPS_EXT.1 FCS_COP.1(1)	CBC (128 and 256 bits)	5242	EXE OpenSSL Cryptographic Module	MPC8377E
SHS (SHA-1, SHA-256, SHA-384)	Cryptographic hashing services FCS_TLSC_EXT.1 FCS_TLSS_EXT.2 FCS_HTTPS_EXT.1 FCS_COP.1(1) FCS_COP.1(3)	Byte Oriented	4220	EXE OpenSSL Cryptographic Module	MPC8377E
DRBG	Deterministic random bit generation services in accordance with ISO/IEC 18031:2011 FCS_TLSC_EXT.1 FCS_TLSS_EXT.2 FCS_HTTPS_EXT.1 FCS_RBG_EXT.1	CTR_DRBG (AES 256)	2004	EXE OpenSSL Cryptographic Module	MPC8377E
ECDSA	Digital Signatures FCS_TLSC_EXT.1 FCS_TLSS_EXT.2 FCS_HTTPS_EXT.1 FCS_COP.1(2)	FIPS PUB 186-4 Key Generation, Public Key Validation (Curves: P-256, P-384, P-521)	1364	EXE OpenSSL Cryptographic Module	MPC8377E
HMAC	Keyed hashing services FPT_TUD_EXT.1 FCS_COP.1(4)	HMAC-SHA-256	3559	EXE OpenSSL Cryptographic Module	MPC8377E
KAS ECC	Ephemeral Key Agreement FCS_CKM.1 FCS_CKM.2	Key Agreement (Initiator, Responder) EC: P-256, SHA-256	1785	EXE OpenSSL Cryptographic Module	MPC8377E

		ED: P-384, SHA-384			
RSA	Signature Verification FCS_CKM.1 FCS_CKM.2 FCS_COP.1(2)	FIPS PUB 186-4 Key Generation (2048-bit key)	2801	EXE OpenSSL Cryptographic Module	MPC8377E

Table 2. CAVP Certificate References

2.3.3 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" MMA10G-EXE users do not access MMA10G-EXE directly; they control IP video switching through the MMA10G-EXE using a switch control system, such as Evertz's Magnum. The switching of those IP video transport stream 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 user name 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. The TOE uses X.509v3 certificates as defined by RFC 5280 to support authentication for TLS/HTTPS connections.

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.

2.3.4 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:

- Administer the TOE locally and remotely
- Configure the access banner
- Configure the cryptographic services
- Update the TOE and verify the updates using digital signature capability prior to installing those updates
- Specify the time limits of session inactivity

All of these management functions are restricted to an Administrator, which covers all administrator roles. Administrators are individuals who manage specific type of administrative tasks. In MMA10G-EXE only the admin role exists, since there is no provision for "regular" users to access MMA10G-EXE directly

(as described above), and the portion of MMA10G-EXE they access and control are outside the scope of the TOE.

Primary management is done using the web-based interface using HTTPS. This provides a network administration console from which one can manage various identity services. These services include authentication, authorization and reporting. All of these services can be managed from the web browser, 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.

2.3.5 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 is used as the time stamp that is applied to TOE generated audit records. An external NTP server can be used for time updates. The TOE also ensures firmware updates are from a reliable source. Finally, the TOE performs testing to verify correct operation.

In order for updates to be installed on the TOE, an administrator initiates the process from the web interface. MMA10G-EXE automatically uses the RSA digital signature mechanism to confirm the integrity of the product before installing the update.

2.3.6 TOE Access

Aside from the automatic Administrators session termination due to inactivity describes 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 interface prior to allowing any administrative access to the TOE.

2.3.7 Trusted Paths/Channels

The TOE allows the establishment of a trusted path between a video control system (such as Evertz' Magnum) and the MMA10G-EXE. The TOE also establishes a secure connection for sending syslog data to a syslog server using TLS and other external authentication stores using TLS-protected communications.

2.4 Excluded Functionality

N/A

3 TOE Description

Evertz MMA10G-EXE is a 10GbE switch designed for video transport over IP. Different chassis are capable of supporting up to 2304 10GbE ports, with a total switching capacity of 46 Tb/s.

MMA10G-EXE's internal frame controllers provide connectivity to remote control panels and 3rd party control devices such as automation systems via Ethernet ports. Using MAGNUM, as the SDVN orchestration and control system, the MMA10G-EXE makes system installations with advanced tie-lines, automated pathfinding, and advanced control surfaces easy to implement and manage. MMA10G-EXE also provides extensive signal monitoring of the line cards, power supply voltages, interior temperatures and fan speeds.

4 Conformance Claims (ASE_CCL)

4.1.1 Common Criteria Claims

The following conformance claims are made for the TOE and ST:

- **CC v3.1, Rev. 3 Conformant**
  The ST and TOE are conformant to Common Criteria version 3.1, Revision 5.

- **Part 2 Extended**
  The ST is Common Criteria Part 2 extended.

- **Part 3 Conformant**
  The ST is Common Criteria Part 3 Conformant.

- **PP Conformant**
  The ST complies to the NDcPP (Collaborative Protection Profile "Security Requirements for Network Devices"), Version 1.0, with additional requirements drawn from Appendix C of the NDcPP.

4.1.2 NIAP Technical Decisions

MMA10G-EXE conforms to the requirements of the following NIAP Technical Decisions:

- 0291: NIT technical decision for DH14 and FCS_CKM.1
- 0290: NIT technical decision for physical interruption of trusted path/channel
- 0289: NIT technical decision for FCS_TLSC_EXT.x.1 Test 5e
- 0281: NIT Technical Decision for Testing both thresholds for SSH rekey
- 0262: NIT Technical Decision for TLS server testing - Empty Certificate Authorities list
- 0257: NIT Technical Decision for Updating FCS_DTLSC_EXT.x.2/FCS_TLSC_EXT.x.2 Tests 1-4
- 0256: NIT Technical Decision for Handling of TLS connections with and without mutual

- 0255: NIT Technical Decision for TLS Server Tests - Issue 3: Verification of application of encryption
- 0235: NIT Technical Decision adding DH group 14 to the selection in FCS_CKM.2
- 0228 – NIT Technical Decision for CA certificates - basicConstraints validation
- 0227 – NIT Technical Decision for TOE acting as a TLS Client and RSA key generation
- 0226 – NIT Technical Decision for TLS Encryption Algorithms
- 0225 – NIT Technical Decision for Make CBC cipher suites optional in IPsec
- 0224 – NIT Technical Decision Making DH Group 14 optional in FCS_IPSEC_EXT.1.11
- 0223 – NIT Technical Decision for "Expected" vs "unexpected" DNs for IPsec Communications
- 0201 – NIT Technical Decision for Use of intermediate CA certificates and certificate hierarchy depth
- 0200 – NIT Technical Decision for Password authentication for SSH clients
- 0199 – NIT Technical Decision for Elliptic Curves for Signatures
- 0195 – NIT Technical Decision Making DH Group 14 optional in FCS_IPSEC_EXT.1.11
- 0191 – NIT Technical Decision for Using secp521r1 for TLS communication
- 0189 – NIT Technical Decision for SSH Server Encryption Algorithms
- 0188 – NIT Technical Decision for Optional use of X.509 certificates for digital signatures
- 0187 – NIT Technical Decision for Clarifying FIA_X509_EXT.1 test 1
- 0186 – NIT Technical Decision for Applicability of X.509 certificate testing to IPsec
- 0185 – NIT Technical Decision for Channel for Secure Update.
- 0184 – NIT Technical Decision for Mandatory use of X.509 certificates
- 0183 – NIT Technical Decision for Use of the Supporting Document
- 0182 – NIT Technical Decision for Handling of X.509 certificates related to ssh-rsa and remote comms.
- 0181 – NIT Technical Decision for Self-testing of integrity of firmware and software.
- 0170 – NIT Technical Decision for SNMPv3 Support
- 0169 – NIT Technical Decision for Compliance to RFC5759 and RFC5280 for using CRLs
- 0168 – NIT Technical Decision for Mandatory requirement for CSR generation
- 0167 – NIT Technical Decision for Testing SSH 2^28 packets
- 0165 – NIT Technical Decision for Sending the ServerKeyExchange message when using RSA
- 0164 – NIT Technical Decision for Negative testing for additional ciphers for SSH
- 0160 – NIT Technical Decision for Transport mode and tunnel mode in IPSEC communications
- 0156 – NIT Technical Decision for SSL/TLS Version Testing in the NDcPP v1.0 and FW cPP v1.0
- 0155 – NIT Technical Decision for TLSS tests using ECDHE in the NDcPP v1.0.

- 0154 – NIT Technical Decision for Versions of TOE Software in the NDcPP v1.0 and FW cPP v1.0
- 0153 – NIT Technical Decision for Auditing of NTP Time Changes in the NDcPP v1.0 and FW cPP v1.0
- 0152 – NIT Technical Decision for Reference identifiers for TLS in the NDcPP v1.0 and FW cPP v1.0
- 0151 – NIT Technical Decision for FCS_TLSS_EXT Testing - Issue 1 in NDcPP v1.0.
- 0150 – NIT Technical Decision for Removal of SSH re-key audit events in the NDcPP v1.0 and FW cPP v1.0
- 0143 – NIT Technical Decision for Failure testing for TLS session establishment in NDcPP and FWcPP
- 0130 – NIT Technical Decision for Requirements for Destruction of Cryptographic Keys
- 0126 – NIT Technical Decision for TLS Mutual Authentication
- 0125 – NIT Technical Decision for Checking validity of peer certificates for HTTPS servers
- 0117 – NIT Technical Decision for FIA_X509_EXT.1.1 Requirement in NDcPP
- 0116 – NIT Technical Decision for a Typo in reference to RSASSA-PKCS1v1_5 in NDcPP and FWcPP
- 0115 – NIT Technical Decision for Transport mode and tunnel mode in IPsec communication in NDcPP and FWcPP
- 0114 – NIT Technical Decision for Re-Use of FIPS test results in NDcPP and FWcPP
- 0113 – NIT Technical Decision for testing and trusted updates in the NDcPP v1.0 and FW cPP v1.0
- 0112 – NIT Technical Decision for TLS testing in the NDcPP v1.0 and FW cPP v1.0.
- 0111 – NIT Technical Decision for third party libraries and FCS_CKM.1 in NDcPP and FWcPP
- 0096 – NIT Technical Interpretation regarding Virtualization
- 0095 – NIT Technical Interpretations regarding audit, random bit generation, and entropy in NDcPP
- 0094 – NIT Technical Decision for validating a published hash in NDcPP
- 0093 – NIT Technical Decision for FIA_X509_EXT.1.1 Requirement in NDcPP
- 0090 – NIT Technical Decision for FMT_SMF.1.1 Requirement in NDcPP

5 Definition of the Security Problem

This section identifies the following:

- Significant assumptions about the TOE's operational environment.
- IT related threats to the organization countered by the TOE.
- Environmental threats requiring controls to provide sufficient protection.
- Organizational security policies for the TOE as appropriate.

This document identifies assumptions as A.assumption with "assumption" specifying a unique name. Threats are identified as T.threat with "threat" specifying a unique name. Organizational Security Policies (OSPs) are identified as P.osp with "osp" specifying a unique name.

The security problem & the associated assumptions, threats, etc are taken directly from the NDcPP v. 1.0.

5.1 Assumptions

The following assumptions about the operational environment are made regarding its ability to provide security functionality. If the TOE is placed in an operational environment that does not meet these assumptions, the TOE may not be able to provide all of its security functionality.

Assumption	Assumption Definition
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 and/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 will not include any requirements on physical tamper protection or other physical attack mitigations. The cPP will 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.
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 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 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 being appropriately trained, following policy, and adhering to guidance documentation. Administrators are trusted to ensure passwords/credentials have sufficient strength and entropy and to lack malicious intent when administering the device. The network device is not expected to be capable of defending against a malicious administrator that actively works to bypass or compromise the security of the device.
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.

Table 3. Operational Environment Assumptions

5.1.1 Threat Model

The table below shows the threats that are to be countered by the TOE, its operational environment, or a combination of the two.

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

T.UPDATE_COMPROMISE	Threat agents may attempt to provide a compromised update of the software or firmware which undermines the security functionality of the device. Non-validated updates or updates validated using non-secure or weak cryptography leave the update firmware vulnerable to surreptitious alteration.
T.UNDETECTED_ACTIVITY	Threat agents may attempt to access, change, and/or modify the security functionality of the network device without administrator awareness. This could result in the attacker finding an avenue (e.g., misconfiguration, flaw in the product) to compromise the device and the administrator would have no knowledge that the device has been compromised.
T.SECURITY_FUNCTIONALITY_COMPROMISE	Threat agents may compromise credentials and device data enabling continued access to the network device and its critical data. The compromise of credentials include replacing existing credentials with an attacker's credentials, modifying existing credentials, or obtaining the administrator or device credentials for use by the attacker.
T.PASSWORD_CRACKING	Threat agents may be able to take advantage of weak administrative passwords to gain privileged access to the device. Having privileged access to the device provides the attacker unfettered access to the network traffic, and may allow them to take advantage of any trust relationships with other network devices.
T.SECURITY_FUNCTIONALITY_FAILURE	A component of the network device may fail during start-up or during operations causing a compromise or failure in the security functionality of the network device, leaving the device susceptible to attackers.

Table 4. Threat Model

5.2 Organizational Security Policies (OSP)

An organizational security policy is a set of rules, practices and procedures imposed by an organization to address its security needs. The table below shows the OSPs that are to be enforced by the TOE, its operational environment or a combination of the two.

Threat Name	**Threat Definition**
P.ACCESS_BANNER	The TOE shall display an initial banner describing restrictions of use, legal agreements or any other appropriate information to which users consent by accessing the TOE.

Table 5. Organizational Security policies

6 Security Objectives (ASE_OBJ)

This section identifies the security objectives of the TOE and the IT Environment. The security objectives identify the responsibilities of the TOE and the TOE's IT environment in meeting the security needs.

6.1 Security Objectives for the TOE

The collaborative Protection Profile for Network Devices v1.0 does not define any security objectives for the TOE

6.2 Security Objectives for the Operational Environment (OE)

All of the assumptions stated in Section 3.1 are considered to be security objectives for the environment. The following are the Protection Profile non-TOE security objectives, which, in addition to those assumptions, are to be satisfied without imposing technical requirements on the TOE. That is, they will not require the implementation of functions in the TOE hardware and/or software. Thus, they will be satisfied largely through application of procedural or administrative measures.

Those objectives are described in the table below:

Environment Security Objective	IT Environment Security Objective Definition
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.
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	TOE Administrators are trusted to follow and apply all administrator guidance in a trusted manner.
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

Table 6. Security Objectives for the Environment

7 Security Requirements

This section specifies the requirements for the TOE. The security functional requirements correspond to the security functions implemented by the TOE, as required by the PP.

7.1 TOE Security Functional Requirements (SFR)

This sub-section specifies the SFRs for the TOE. It organizes the SFRs by CC classes as per the table below.

CC Functional		Security Functional Requirements	
Class	**Description**	**TOE SFR**	**Description**
FAU	Security Audit	FAU_GEN.1	Audit Data Generation
		FAU_GEN.2	User Identity Association
		FAU_STG_EXT.1	External Audit Trail Storage
FCS	Cryptographic Support	FCS_CKM.1	Cryptographic Key Generation
		FCS_CKM.2	Cryptographic Key Establishment
		FCS_CKM.4	Cryptographic Key Destruction
		FCS_COP.1(1)	Cryptographic Operation (AES Data Encryption/Decryption)
		FCS_COP.1(2)	**Cryptographic Operation (Signature Generation and Verification)**
		FCS_COP.1(3)	Cryptographic Operation (Hash Algorithm)
		FCS_COP.1(4)	Cryptographic Operation (Keyed-Hash Algorithm)
		FCS_RBG_EXT.1	Random Bit Generation
		FCS_TLSC_EXT.1	Explicit: TLS (Client)
		FCS_TLSS_EXT.2	Explicit: TLS (Server) with mutual authentication
		FCS_HTTPS_EXT.1	Explicit: HTTPS
FIA	Identification and Authentication	FIA_PMG_EXT.1	Password Management
		FIA_UIA_EXT.1	User Identification and Authentication
		FIA_UAU_EXT.2	Password-Based Authentication Mechanism
		FIA_UAU.7	Protected Authentication Feedback
		FIA_X509_EXT.1	X.509 Certificate Validation
		FIA_X509_EXT.2	X.509 Certificate Authentication
		FIA_X509_EXT.3	X.509 Certificate Requests
FMT	Security Management	FMT_MOF.1(1) / Trusted Update	Management of security functions behavior
		FMT_MTD.1	Management of TSF Data
		FMT_SMF.1	Specification of Management Functions
		FMT_SMR.2	Restrictions on Security Roles
FPT	Protection of the TSF	FPT_SKP_EXT.1	Protection of TSF Data (for Reading of All Symmetric Keys)
		FPT_APW_EXT.1	Protection of Administrator Passwords
		FPT_TST_EXT.1	TSF Testing
		FPT_TUD_EXT.1	Trusted Update
		FPT_STM.1	Reliable Time Stamps
FTA	TOE Access	FTA_SSL_EXT.1	TSF-Initiated Session Locking
		FTA_SSL.3	TSF-Initiated Termination
		FTA_SSL.4	User-Initiated Termination
		FTA_TAB.1	Default TOE Access Banners
FTP	Trusted Path/Channels	FTP_ITC.1	Inter-TSF Trusted Channel
		FTP_TRP.1	Trusted Path

Table 7. TOE Security Functional Requirements

7.1.1 Security Audit (FAU)

7.1.1.1 FAU.GEN.1 – Audit Data Generation

FAU_GEN1.1 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) All auditable events for the not specified level of audit; and
c) *All administrative actions comprising:*
   - *Administrative login and logout (name of user account shall be logged if individual user accounts are required for administrators).*
   - *Security related configuration changes (in addition to the information that a change occurred it shall be logged what has been changed).*
   - *Generating import of, changing, or deleting of cryptographic keys (in addition to the action itself a unique key name or key reference shall be logged).*
   - *Resetting passwords (name of related user account shall be logged).*
   - *Starting and stopping services (if applicable)*
   - No other actions
d) *Specifically defined auditable events listed in **Table 8.** TOE Security Functional Requirements and Auditable Events*
e) .

SFR	Auditable Events	Additional Audit Record Contents
FAU_GEN.1	None	
FAU_GEN.2	None	
FAU_STG_EXT.1	None	
FCS_CKM.1	None	
FCS_COP.1(1)	None	
FCS_COP.1(2)	None	
FCS_COP.1(3)	None	
FCS_COP.1(4)	None	
FCS_RBG_EXT.1	None	
FDP_RIP.2	None	
FCS_TLSC_EXT.1	Failure to Establish a TLS Session (Client).	Non-TOE Endpoint (IP Address)
		Reason for Failure
	Establishment of a TLS Session (Client)	Non-TOE Endpoint (IP Address)
	Termination of a TLS Session (Client)	Non-TOE Endpoint (IP Address)
FCS_TLSS_EXT.2	Failure to Establish a TLS Session (Server).	Non-TOE Endpoint (IP Address)
		Reason for Failure
	Establishment of a TLS Session (Server)	Non-TOE Endpoint (IP Address)

SFR	Auditable Events	Additional Audit Record Contents
	Termination of a TLS Session (Server)	Non-TOE Endpoint (IP Address)
FCS_HTTPS_EXT.1	Failure to Establish an HTTPS Session.	Non-TOE Endpoint (IP Address)
		Reason for Failure
	Establishment of an HTTPS Session	Non-TOE Endpoint (IP Address)
	Termination of an HTTPS Session	Non-TOE Endpoint (IP Address)
FIA_PMG_EXT.1	None	
FIA_UIA_EXT.1	All Use of the Identification and Authentication Mechanism	Provided User Identity
		Origin of Attempt (IP Address, etc.)
FIA_UAU_EXT.2	All Use of the Identification and Authentication Mechanism	Origin of Attempt (IP Address, etc.)
FIA_UAU.7	None	
FIA_X509.EXT.1	Unsuccessful attempt to validate a certificate	Reason for failure
FIA_X509.EXT.2	None	None
FIA_X509.EXT.3	None	None
FMT_MOF.1(1) / TrustedUpdate	Any attempt to initiate a manual update.	None.
FMT_MTD.1	All management activities of TSF data.	None
FMT_SMF.1	None	None
FMT_SMR.2	None	None
FPT_SKP_EXT.1	None	None
FPT_APW_EXT.1	None	None
FPT_TST_EXT.1	None	None
FPT_TUD_EXT.1	Initiation of Update; result of the update attempt	No additional information.
FPT_STM.1	Changes to Time	Old Time Value
		New Time Value
		Origin of Attempt (IP Address, etc.)
FTA_SSL_EXT.1	Any Attempts at Unlocking an Interactive Session	None
FTA_SSL.3	Termination of a Remote Session by the Session Locking Mechanism	None
FTA_SSL.4	Termination of an Interactive Session	None
FTA_TAB.1	None	None
FTP_ITC.1	Initiation of a Trusted Channel	Identification of the Initiator
		Identification of the Target
	Termination of a Trusted Channel	Identification of the Initiator
		Identification of the Target
	Failure of Trusted Channel Functions	Identification of the Initiator

SFR	Auditable Events	Additional Audit Record Contents
		Identification of the Target
FTP_TRP.1	Initiation of a Trusted Path	Identification of Claimed User Identity
	Termination of a Trusted Path	Identification of Claimed User Identity
	Failure of Trusted Path Functions	Identification of Claimed User Identity

Table 8. TOE Security Functional Requirements and Auditable Events

FAU_GEN.1.2 The TSF shall record within each audit record at least the following information:

a) Date and time of the event, type of event, subject identity, and the outcome (success or failure) of the event; and
b) For each audit type, based on the auditable event definitions of the functional components listed in the PP/ST, *information specified in column three of Table 8. TOE Security Functional Requirements and Auditable Events*
c) above.

7.1.1.2 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.

7.1.1.3 FAU_STG_EXT.1 – External Audit Trail Storage

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

NOTE: For the purposes of this ST, the external IT entity is an organizationally-provided syslog server.

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

FAU_STG_EXT.1.3 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.

7.1.2 Cryptographic Support (FCS)

7.1.2.1 FCS_CKM.1 – Cryptographic Key Generation (Refined)

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

- ***RSA schemes using cryptographic key sizes of 2048-bit or greater that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.3;***
- ***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***

7.1.2.2 FCS_CKM.2 – Cryptographic Key Establishment (Refined)

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 meets the following: NIST Special Publication 800-56B Revision 1, "Recommendation for Pair-Wise Key Establishment Schemes Using Integer Factorization Cryptography";***
- ***Elliptic curve-based key establishment schemes that meets the following: NIST Special Publication 800-56A Revision 2, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography"***

7.1.2.3 FCS_CKM.4 – Cryptographic Key Zeroization

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;*
  - *instructs a part of the TSF to destroy the abstraction that represents the key*

that meets the following standard: ***No Standard*.**

7.1.2.4 FCS_COP.1(1) – Cryptographic Operation (AES Data Encryption/Decryption)

FCS_COP.1.1(1) The TSF shall perform *encryption/decryption* in accordance with a specified cryptographic algorithm *AES used in* CBC *mode* and cryptographic key sizes 128-bits and 256-bits that meet the following: *AES as specified in ISO 18033-3,* CBC as specified in ISO 10116.

7.1.2.5 FCS_COP.1(2) – Cryptographic Operation (Signature Generation and Verification)

FCS_COP.1.1(2) The TSF shall perform *cryptographic signature services (generation and verification)* in accordance with a specified cryptographic algorithm:

- *RSA Digital Signature Algorithm and cryptographic key sizes (modulus) 2048 bits.*
- *Elliptic Curve Digital Signature Algorithm and cryptographic key sizes 256 bits*

that meet the following:

- *For RSA schemes: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 5.5, using PKCS#1 v2.1 Signature Schemes RSASSA-PSS and/or RSASSA-PKCS1v1_5; ISO/IEC 9796-2, Digital Signature scheme 2 or Digital Signature scheme 3*
- *For ECDSA schemes: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 6 and Appendix D, Implementing "NIST curves" P-256, P-384, P-521; ISO/IEC 14888-3, Section 6.4*

7.1.2.6 FCS_COP.1(3) – Cryptographic Operation (Hash Algorithm)

FCS_COP.1.1(3) The TSF shall perform *cryptographic hashing services* in accordance with a specified cryptographic algorithm **(**SHA-1, SHA-256, SHA-384**)** that meet the following: *ISO/IEC 10118-3:2004*.

7.1.2.7 FCS_COP.1(4) – Cryptographic Operation (Keyed Hash Algorithm)

FCS_COP.1.1(4) The TSF shall perform *keyed-hash message authentication* in accordance with a specified cryptographic algorithm HMAC-SHA-256 and cryptographic key sizes 256 bits **and message digest size 256 bits**, that meet the following: ISO/IEC 9797-2:2011, Section 7 "MAC Algorithm 2".

7.1.2.8 FCS_RGB_EXT.1 –Random Bit Generation

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

FCS_RBG_EXT.1.2 The deterministic RBG shall be seeded by at least one entropy source that accumulates entropy from *one* software-based noise source with a minimum of 256 bits of entropy at least equal to the greatest security strength, according ISO/IEC 18031:2011 Table C.1 "Security Strength Table for Hash Functions," of keys and hashes that it will generate.

7.1.3 Identification and Authorization (FIA)

7.1.3.1 FIA_PMG_EXT.1 – Password Management

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

- *Passwords shall be able to be composed of any combination of upper and lower case letters, numbers, and the following special characters: "!"; "@"; "#"; "$"; "%"; "^"; "&"; "*"; "("; ")";"~"; "`"; "_"; "-"; "+"; "="; "{"; "["; "}"; "]"; "|"; "\"; ":"; ";"; ["]; ']; "<"; ","; ">"; "."; "?"; "/"; " ";*
- *Minimum password length shall be settable by the Security Administrator, and support passwords of 15 characters or greater.*

7.1.3.2 FIA_UIA_EXT.1 – User Identification and Authorization

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

7.1.3.3 FIA_UAU_EXT.2 – Extended: Password-Based Authentication Mechanism

FIA_UAU_EXT.2.1 The TSF shall provide a local password-based authentication mechanism, *none* to perform administrative user authentication.

7.1.3.4 FIA_UAU.7 – Protected Authentication Feedback

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

7.1.4 Security Management (FMT)

7.1.4.1 FMT_MOF.1(1)/TrustedUpdate – Management of Security Functions Behavior

FMT_MOF.1.1(1)/TrustedUpdate The TSF shall restrict the ability to enable the functions *to perform manual update* to *Security Administrators*.

7.1.4.2 FMT_MTD.1 – Management of TSF Data (for General TSF Data)

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

7.1.4.3 FMT_SMF.1 – Specification of Management Functions

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

- *Ability to administer the TOE locally and remotely;*
- *Ability to configure the access banner;*
- *Ability to configure the session inactivity time before session termination or locking;*
- *Ability to update the TOE, and to verify the updates using digital signature capability prior to installing those updates*
- ***No other capabilities.***

7.1.4.4 FMT_SMR.2 – Restrictions on Security Roles

FMT_SMR.2.1 The TSF shall maintain the roles:

- *Security Administrator*.

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

FMT_SMR.2.3 The TSF shall ensure that the conditions

- *The Security Administrator role shall be able to administer the TOE locally;*
- *The Security Administrator role shall be able to administer the TOE remotely;*

are satisfied.

7.1.5 Protection of the TSF (FPT)

7.1.5.1 FPT_SKP_EXT.1 – Extended: 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 key and private keys.

7.1.5.2 FPT_APW_EXT.1 – Extended: Protection of Security Administrator Passwords

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

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

7.1.5.3 FPT_TST_EXT.1 – TSF Testing

FPT_TST_EXT.1.1 The TSF shall run a suite of the following self tests during initial start-up (on power on) to demonstrate the correct operation of the TSF:

- *firmware integrity check that compares the SHA256 checksum of the loaded firmware with a permanently stored hash value;*
- *from the previous successful upgrade, or in the case of first time upgrade, the one-time user-generated hash value;*
- *Presence of certificate and public key files.*

7.1.5.4 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 a means to authenticate firmware/software updates to the TOE using a digital signature mechanism prior to installing those updates.

7.1.5.5 FPT_STM.1 – Reliable Time Stamps

FPT_STM.1.1 The TSF shall be able to provide reliable time stamps.

7.1.6 TOE Access (FTA)

7.1.6.1 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.

7.1.7 FTA_SSL.3 – TSF-Initiated Termination

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

7.1.7.1 FTA_SSL.4 – User-Initiated Termination

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

7.1.7.2 FTA_TAB.1 – Default TOE Access Banners

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

7.1.8 Trusted Path/Channels (FTP)

7.1.8.1 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 **authorized IT entities supporting the following capabilities:**

- **Video Switch Control System (such as Evertz' Magnum)**
- **Audit Server**

that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from disclosure and detection of modification of the channel data.

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

FTP_ITC.1.3 The TSF shall initiate communication via the trusted channel for *auditing services and system logging.*

7.1.8.2 FTP_TRP.1 – Trusted Path

FTP_TRP.1.1 The TSF shall be **capable of using** TLS, HTTPS **to** provide a communication path between itself and **authorized remote administrators** that is logically distinct from other communications 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 Refinement: The TSF shall permit **remote administrators** to initiate communication via the trusted path.

FTP_TRP.1.3 Refinement: The TSF shall require the use of the trusted path for **initial administrator authentication and all remote administration actions**.

7.2 Selection-Based Requirements (Annex B)

MMA10G-EXE performs TLS server-side tasks when communicating with external control interfaces (web browser, dedicated touch panel). When communicating via the web, HTTPS is required. Therefore, FCS_HTTPS and FCS_TLSS applies for these functions.

MMA10G-EXE serves as the client for TLS-based syslog communication. Therefore, FCS_TLSC applies for this function.

7.2.1 Cryptographic Support (FCS)

7.2.1.1 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 using TLS.

FCS_HTTPS_EXT.1.3 The TSF shall establish the connection only if the peer presents a valid certificate during handshake, or the peer initiates handshake.

7.2.1.2 FCS_TLSC_EXT.1– TLS Client Protocol

FCS_TLSC_EXT.1.1 The TSF shall implement TLS1.2 (RFC 5246) supporting the following ciphersuites:

TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC3268
TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC3268
TLS_RSA_WITH_AES_256_CBC_SHA as defined in RFC3268
TLS_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC3268
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289

FCS_TLSC_EXT.1.2 The TSF shall verify that the presented identifier matches the reference identifier according to RFC 6125.

FCS_TLSC_EXT.1.3 The TSF shall only establish a trusted channel if the peer certificate is valid.

FCS_TLSC_EXT.1.4 The TSF shall present the Supported Elliptic Curves Extension in the Client Hello with the following NIST curves: secp256r1, secp384r1, secp521r1 and no other curves.

7.2.1.3 FCS_TLSS_EXT.2 – TLS Server Protocol with mutual authentication

FCS_TLSS_EXT.2.1 The TSF shall implement TLS1.2 (RFC 5246) supporting the following ciphersuites:

TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC3268

TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC3268
TLS_RSA_WITH_AES_256_CBC_SHA as defined in RFC3268
TLS_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC3268
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289

FCS_TLSS_EXT.2.2 The TSF shall deny connections from clients requesting SSL 2.0, SSL 3.0, TLS 1.0 and TLS 1.1.

FCS_TLSS_EXT.2.3 The TSF shall perform RSA key establishment with key size 2048 bits; generate EC Diffie-Hellman parameters over NIST curves secp256r1, secp384r1, secp521r1 and no other curves.

FCS_TLSS_EXT.2.4 The TSF shall support mutual authentication of TLS clients using X.509v3 certificates.

FCS_TLSS_EXT.2.5 The TSF shall not establish a trusted channel is the peer certificate is invalid.

FCS_TLSS_EXT.2.6 The TSF shall not establish a trusted channel if the distinguished name (DN) or Subject Alternative Name (SAN) contained in a certificate does not match the expected identifier for the peer.

7.2.1.4 FIA_X509_EXT.1 – X.509 Certificate Validation

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

- RFC 5280 certificate validation and certificate path validation
- The certificate path must terminate with a trusted CA certificate.
- The TSF shall validate a certificate path by ensuring the presence of the basicConstraints extension and that the CA flag is set to TRUE for all CA certificates.
- The TSF shall validate the revocation status of the certificate using a Certificate Revocation List (CRL) as specified in RFC 5280 Section 6.3**.**
- The TSF shall validate the extendedKeyUsage field according to the following rules:
  - *Certificates used for trusted updates and executable code integrity verification shall have the Code Signing purpose (id-kp 3 with OID 1.3.6.1.5.5.7.3.3) in the extendedKeyUsage field.*
  - *Server certificates presented for TLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field.*
  - *Client certificates presented for TLS shall have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the extendedKeyUsage field.*
  - *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 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.

7.2.1.5 FIA_X509_EXT.2 – X.509 Certificate Authentication

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

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

7.2.1.6 FIA_X509_EXT.3 – X.509 Certificate Requests

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

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

8 TOE Security Assurance Requirements

The TOE meets the security assurance requirements of NDPP v1.0. The following table is the summary of those requirements:

Assurance Class		Assurance Components	
Class	**Description**	**Component**	**Description**
ADV	Development	ADV_FSP.1	Basic Functional Specification
AGD	Guidance Documents	AGD_OPE.1	Operational User Guidance
		AGD_PRE.1	Preparative User Guidance
ATE	Tests	ATE_IND.1	Independence Testing – Conformance
AVA	Vulnerability Assessment	AVA_VAN.1	Vulnerability Analysis
ALC	Life Cycle Support	ALC_CMC.1	Labeling of the TOE
		ALC_CMS.1	TOE CM Coverage

Table 9. TOE Security Assurance Requirements

8.1 TOE Summary Specification (TSS)

Requirement	Rationale
FAU_GEN.1 FAU_GEN.2 FAU_STG_EXT.1	MMA10G-EXE 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) and a description of the event. Audit records are created when an auditable event that belongs to a set of predefined events had occurred. The set of auditable events can be sub-categorized into functional events and access events. MMA10G-EXE stores audit logs internally. The internal logs are stored unencrypted, but they are only accessible (and then read-only) via the web browser, which can only be used by Administrators. Information is also sent (using TLS 1.2) to an external Syslog server. 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 be also uploaded to MMA10G-EXE. The TOE overwrites previous audit records on a circular (FIFO) basis when the local storage space for audit data is full.
FCS_CKM.1 FCS_CKM.2 FCS_CKM.4	The cryptographic module protects the management interfaces (TLS/HTTPS). It uses the following cryptographic algorithms: AES128 & 256 (Symmetric Cipher), RSA with 2048 bit keys (Asymmetric Cipher), SHA256 (Hashed MAC), RSA with 2048 bit keys (Digital Signatures), X509 (Certificate Encoding) and DRBG-AES-256-CTR Mode (Random number generation). MMA10G-EXE uses the EXE OpenSSL Cryptographic Module. The TOE supports 2048-bit RSA keys. The TOE acts as both sender and recipient for RSA-based key establishment schemes. The underlying platform provides key confirmation services. Elliptic Curve Diffie Hellman ephemeral key exchange is supported with curves P-256, P-384, and P-521. The TOE acts as both sender and recipient for ECDHE key establishment schemes. 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. Cryptographic keys are destroyed by first overwriting the key file content with all 0s. This will be done three times. Then a read-verification will be performed to ensure that the entire content has really been changed to zeros and not any other values. If this steps fails, then the file will be over-written again 3 times with 0s until the read-verify step succeeds. The following keys are stored: • the trust CA certificate, which is used for certificate verification; • the server certificate, which is used for HTTP web service and Magnum TLS connection; • the private key matching the server certificate, which is used for de-encryption; • encrypted credential file, which is used for web service login; • CRL (certificate revocation list) file, which is used for certificate verification; All of these cryptographic keys are stored in plaintext on nonvolatile NOR flash storage. No direct interface/access is provided to view or modify the contents of these files.
FCS_COP.1(1)	The secure version of MMA10G-EXE is not configurable WRT cryptographic operation. The system defaults to the selected cryptographic modes and is not alterable when the system is placed into High Security mode.
FCS_COP.1(2)	Digital signatures are used during TLS connection establishment and verification of trusted updates.
FCS_COP.1(3)	The TOE implements SHA-1/SHA256/SHA384 hashing in byte-oriented mode. Hashes are used for TLS, firmware integrity check during power-on-self-test and upgrade, and password verification.
FCS_COP.1(4)	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 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 MMA10G-EXE, 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. HMAC-SHA-256 is the only keyed-hash message authentication function used by MMA10G-EXE. The following parameters are supported for HMAC functions within the TOE: • Hash: SHA-256 • Key Length: 512 bits • Block size: 512 bits • MAC Length: 256 bits
FCS_RBG_EXT.1	As determined in *Evertz Microsystems MMA10G-EXE Entropy Assessment Report*, the Linux kernel on which the MMA10G-EXE application is built uses */dev/random* as the entropy source for all random numbers. The functions which obtain random numbers from the RBG are: • Haveged • Linux Kernel Entropy Please see the *Evertz Microsystems Entropy Assessment Report* for MMA10G-EXE for further information, such as seeding parameters.
FCS_HTTPS_EXT.1	MMA10G-EXE uses Apache web server's HTTPS implementation to provide a secure interactive webpage interface for remote administrative functions, and to support secure exchange of user authentication parameters during login. HTTPS uses TLS to securely establish the encrypted session. The sessions are not established if peer's certificates can't be validated. Certificates (MMA10G-EXE's own certificate, trusted CA certificate) can be uploaded on MMA10G-EXE prior to establishing connection with peers. These certificates are used in the TLS handshaking process and is taken care of by TLS protocol implementation.
FCS_TLSC_EXT.1 FCS_TLSS_EXT.2	TLS is used to export audit records and communicate with video switch control systems (e.g. Magnum devices), as well as supporting remote administrator connections. Server-side and client-side TLS work the same way, except that on the server side mutual authentication is supported. MMA10G-EXE specifies only a restricted set of cipher suites that it supports during the negotiation phase with its peer. If no match of

	cipher suites can be found with peer, TLS session will not be started. The following cipher suites are supported: • TLS_RSA_WITH_AES_128_CBC_SHA • TLS_RSA_WITH_AES_128_CBC_SHA256 • TLS_RSA_WITH_AES_256_CBC_SHA • TLS_RSA_WITH_AES_256_CBC_SHA256 • TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 Protocols that do not conform to TLS1.2 are explicitly excluded in MMA10G-EXE's cipher suites MMA10G-EXE only supports cipher suites that use RSA keys or ECDHE/ECDSA for key exchange and authentication. RSA keys are generated with OpenSSL's RSA command line utility. ECDHE supports the curves secp256r1, secp384r1, and secp521r1. MMA10G-EXE uses CRL (certification revocation list) to check for invalid certificates. CRL files which are signed by trusted CA certificated can be imported to MMA10G-EXE. This CRL file will be used by MMA10G-EXE during certificate validation process to check for revocation status of the peer certificates. MMA10G-EXE allows configuration of reference identifier from a peer it expects to connect with before a connection is made. The reference identifier can be any string up to 64 bytes that is present in the peer certificate's DN/SAN field. The verification against DN/SAN peer certificate is implemented within OpenSSL. MMA10G-EXE 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.
FIA_PMG_EXT.1 FIA_UAU_EXT.2	Out of the factory, MMA10G-EXE is configured to use a default password. Any user is then required to update their passwords when they login for the first time. MMA10G-EXE enforces that passwords must meet minimum requirements (length, mix of number of lower/upper case letters, numbers as well as special characters, no common dictionary words. etc). Valid passwords are stored as hashed values.

FIA_UIA_EXT.1	Warning banner is displayed before login prompt becomes ready to accept login credentials from user. Users must acknowledge the warning banner before they can login to the system Authentication of administrator is based on username/password. Prior to successful login, no interface is exposed to allow unauthorized access.
FIA_UAU.7	On the webpage, solid dots are used when entering a password. On serial port access, no feedback of any sort is used.
FIA_X509_EXT.1 FIA_X509_EXT.2 FIA_X509_EXT.3	MMA10G-EXE uses OpenSSL for X.509 certificate validation. The certificate path is validated by ensuring that all the CA certificates have the basicConstraints extension 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. They are not used for any purposes other than establishing TLS sessions. If certificates are uploaded to MMA10G-EXE for its own use those certificates are checked upon upload. Certificates presented by remote TLS servers or by clients using mutual authentication are validated during the establishment of a TLS connection. Each certificate presented as part of a TLS connection with the TOE is checked against each of the installed certificate chains. For an expired certificate, MMA10G-EXE will deny the connection. MMA10G-EXE also uses CRLs to verify whether the certificate or intermediate CA certificate has been revoked. During session establishment with MMA10G-EXE, modifications in the certificate will lead to the failure of connection. If the TOE is unable to reach a CRL, the connection will be accepted. Instructions about generating/downloading CSR and loading certificate can be found on MMA10G-EXE manual. The CSR request includes the Common Name, Organization, Organizational Unit, and Country. The Administrator can only upload one certificate and one CA certificate. The same certificate will be used by MMA10G-EXE for both web service and Magnum control. The same CA will be used for certificate verification. MMA10G-EXE enforces mutual authentication on connections to a Magnum controller. If certificate verification fails, the connection attempt fails and the trusted channel is not established. 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. If the TOE is unable to reach a CRL, the connection will be accepted. This is applicable to all TLS connections.

FMT_MOF.1(1)/TrustedUpdate FMT_MTD.1 FMT_SMF.1 FMT_SMR.2	MMA10G-EXE 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. MMA10G-EXE ensures that only secure values are accepted for security attributes. A Security Administrator can change passwords, and can add, edit and/or delete Security Administrator accounts. The (non-administrative) User has no direct access or control over MMA10G-EXE; a (non-administrative) User may only access an MMA10G-EXE card through Magnum. No administrative functionality is available prior to login.
FPT_SKP_EXT.1 FPT_APW_EXT.1	MMA10G-EXE prevents the unauthorized modification of TSF data. This protection includes self-tests to ensure the correct operation of cryptographic functions. Firmware upgrades (only performed by a Security Administrator) are impossible unless the new firmware first passes two separate authentication tests. MMA10G-EXE relies on trusted channels to protect communications between itself and other trusted services, such as syslog. Communications between the MMA10G-EXE and a remote administrative user are protected via a trusted path. Cryptographic keys are stored in a directory in flash memory. As there is no command line access, users cannot gain any direct access to these files. No passwords are stored in plaintext. Their hashed values are stored instead 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.
FPT_TST_EXT.1	Upon enabling FIPS mode (or on power-up with FIPS mode set "on") the algorithm self-tests required by FIPS are performed. After loading the image, a hash value is computed from the memory partition containing the image. This hash values 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.
FPT_TUD_EXT.1	The site administrators do not have access to install any application. The MMA10G-EXE embedded system can only be updated with the valid firmware release from Evertz. Operators may verify the current version with Web GUI. The current firmware version is displayed on both webpage and in serial console menu.

	Digital delivery of new MMA10G-EXE firmware may be provided via File Transfer Protocol Secure (FTPS) using signed and hashed code. Firmware updates are done from the MMA10G-EXE webpage interface under “upgrade”. During a firmware upgrade, MMA10G-EXE will first verify the HMAC of new firmware code with local stored public key. There’s no way to change the local stored public key by administrators. When HMAC verification passes, MMA10G-EXE will verify the firmware binary header with Evertz-defined proprietary format. If there’s no mismatch, the new firmware code will overwrite the current one. A hashed-value of the images is generated and then signed with Evertz’s private key. The result file (signature) is included in the firmware package together with the actual firmware binary. During upgrade, the signature file is first decrypted using the public key stored on MMA10G-EXE, then the hashed value is re-calculated from the uploaded image binary file and then compared with the decrypted hash value. These hashes must match for this validation to succeed.
FPT_STM.1	Timestamps found in auditable log events use system clock on MMA10G-EXE. Administrators can set the system time clock through serial port console menu after each card reboot. Timestamps found in auditable log events come from system time on MMA10G-EXE. The system time of MMA10G-EXE can only be set through serial port console menu by administrator. 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 MMA10G-EXE is powered off. During MMA10G-EXE system startup, system time is initialized to the time from the hardware clock.
FTA_SSL_EXT.1 FTA_SSL.3 FTA_SSL.4 FTA_TAB.1	Security Administrators can configure a maximum allowable period of inactivity for a Security Administrator session via the console or HTTPS. If there is no user interaction with the MMA10G-EXE for the specified amount of time, the session is terminated. The initial, default session timeout is 15 minutes. Security Administrators may also terminate their own sessions. The MMA10G-EXE also provides for a login banner message to be displayed by the management interfaces (WebEasy or Magnum), to advise Security Administrators regarding the appropriate use of the MMA10G-EXE, and the penalty for its misuse.
FTP_ITC.1	MMA10G-EXE sets up trusted channels with Magnum and syslog servers through the TLS protocol. Specifically, the handshaking

	process must occur and succeed before application level communication could occur. Furthermore, once session is established, TLS ensures the confidentiality, integrity and authenticity of the communication data.
FTP_TRP.1	MMA10G-EXE sets up trusted path with administrators over secure HTTPS session, which again uses TLS as the underlying security protocol to protect communications. TLS handshake needs to be performed prior to HTTPS session establishment, which ensures communications only happen with trusted parties. TLS also ensures the confidentiality, integrity and authenticity of the communication data.

Appendix A. Glossary of Terms

TERM	DEFINITION
AES	Advanced Encryption Standard
AV	Audio-Video, Audiovisual
CBC	Cipher Block Chain
CC	Common Criteria
CO	Cryptography Officer
CTR	Counter (mode)
CWDM	Coarse Wave Division Multiplexing
DFB	Distributed Feedback
DHE	Diffie-Hellman Exchange
DNS	Domain Name Service
DRBG	Deterministic Random Bit Generator
DVI	Digital Video Interface
DWDM	Dense Wave Division Multiplexing
ECDHE	Elliptic Curve Diffie-Hellman Exchange
ECDSA	Elliptic Curve Digital Signature Algorithm
EMX	Evertz Modular Crosspoint
Gb	Gigabit
GCM	Galois/Counter Mode
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IP	Internet Protocol
km	Kilometer(s)
max	Maximum
NDPP	Network Device Protection Profile
nm	Nanometer(s)

NTP	Network Time Protocol
OE	Operational Environment
OOBM	Out of Band Management
RBAC	Role Based Access Control
RFC	Request For Comment
RJ-45	Radio Jack (45)
RS-232	Recommended Standard 232
RSA	Rivest-Shamir-Adelman
SDI	Serial Digital Interface
SFP	Small Form-Factor Pluggable
SFR	Security Functional Requirements
SHA	Secure Hash Algorithm
SMF	Single Mode Fiber
SNMP	Simple Network Management Protocol
SSH	Secure Shell
ST	Security Target
TLS	Transport Layer Security
TOE	Target of Evaluation
TSF	Target Security Function
USB	Universal Serial Bus
VGA	Video Graphics Array