SECURITY TARGET

GMV GNSS CRYPTOGRAPHIC MODULE

Prepared by: Jorge Juan Tejero
Esther Godoy Alves
Alberto Sendino Aragonés
Fernando Elena Benavente
Teresa Feu Basilio
Enrique Robles Uriel

Date: 28/08/24
Version: v2.1.7

TABLE OF CONTENTS

LIST OF TABLES AND FIGURES

1. INTRODUCTION

1.1. PURPOSE

This document consists of the Security Target for the Common Criteria evaluation of the GMV GNSS Cryptographic Module.

1.2. DEFINITIONS AND ACRONYMS

1.2.1. DEFINITIONS

Concepts and terms used in this document and needing a definition are included in the following table:

Concept / Term	Definition
Administrator	Entity that has a level of trust with respect to all policies implemented by the TOE security functionality (TSF)
Protection Profile (PP)	Implementation-independent statement of security needs for a target of evaluation (TOE) type
Role	Pre-defined set of rules establishing the allowed interactions between a user and the target of evaluation (TOE)
Security Objective	Statement of an intent to counter identified threats and/or satisfy identified organization security policies and/or assumptions.
Security Problem Definition	Statement, which in a formal manner defines the nature and scope of the security that the target of evaluation (TOE) is intended to address.
Security Requirement	Requirement, which is part of a target of evaluation (TOE) security specification as defined in a specific security target (ST) or in a protection profile (PP).
Security Target (ST)	Implementation-dependent statement of security requirements for a target of evaluation (TOE) based on a security problem definition.
Target Of Evaluation (TOE)	Set of software, firmware and/or hardware possibly accompanied by guidance, which is the subject of an evaluation.
Threat Agent	Entity that has potential to exercise adverse actions on assets protected by the target of evaluation (TOE).
TOE security functionality (TSF)	Combined functionality of all hardware, software, and firmware of a target of evaluation (TOE) that is relied upon for the correct enforcement of the security functional requirements (SFRs).

Table 1 Definitions

1.2.2. ACRONYMS

Acronyms used in this document and needing a definition are included in the following table:

Acronym	Definition
API	Application Programming Interface
CAVS	Cryptographic Algorithm Validation Test
EAL	Evaluation assurance Level
ESA	European Space agency
FIPS	Federal Information Processing Standard
OSNMA	Open Service Navigation Message Authentication
PP	Protection Profile
RBCA	Role Based Control Access
SFR	Security Functional Requirement
SFP	Security Function Policies
ST	Security Target
TOE	Target of Evaluation
TSF	TOE security functionality
GNSS	Global Navigation Satellite System

Table 2 Acronyms

2. REFERENCES

2.1. REFERENCE DOCUMENTS

The following documents are referenced in this document in the form [RD.x – vx.y.z]:

Ref.	Title	Version	Revision	Date
RD.1 - v2.1.7	Common Criteria for information Technology Security Evaluation. Part 1: introduction and general model	3.1	5	April 2017
RD.2 - v2.1.7	Common Criteria for information Technology Security Evaluation. Part 2: Security functional requirements	3.1	5	April 2017
RD.3 - v2.1.7	Common Criteria for information Technology Security Evaluation. Part 3: Security assurance requirements	3.1	5	April 2017
RD.13 - v2.1.7	Galileo Open Service Navigation Message Authentication (OSNMA) Receiver Guidelines	1.3	-	January 2024
RD.22 - v2.1.7	GMV GNSS Cryptographic Module - Master Project References Document	v2.1.7	-	28/08/24

Table 3 Reference Documents

3. SECURITY TARGET INTRODUCTION

3.1. ST REFERENCE

ST Title	GMV GNSS Cryptographic Module - Security Target
ST Version	v2.1.7
Date	28/08/24
Author	Esther Godoy, Jorge Juan Tejero, Alberto Sendino, Fernando Elena, Teresa Feu, Enrique Robles.

3.2. TOE REFERENCE

TOE developer	GMV Aerospace and Defence
TOE version	v2.1.7
Date	28/08/24
TOE name	GMV GNSS Cryptographic Module*

**The product name may also be referred to as GMV GNSS Module in short. The document may refer to the product simply as GMV GNSS Module, module, or the TOE.*

3.3. TOE OVERVIEW

The TOE specified in the following document is the GMV GNSS Cryptographic Module.

The GMV GNSS Module consists in a software library develop in Linux platform which main purpose is to provide the cryptographic services which will be used in the future for a lot of projects of GNSS, for example the Open Service Navigation Message Authentication (OSNMA) protocol used in Galileo. These services use cryptographic algorithms approved by the European Space agency (ESA) RD.13 - v2.1.7.

The TOE is intended to be implemented in a platform environment that uses **Ubuntu 22.04** or above and includes the **GNU Multiple Precision Arithmetic Library** dependencies. The TOE provides the cryptographic primitives to do the cryptographic functions through an Application Program Interface (API). The connection between the application and the GMV GNSS Module implements network sockets that use TCP/IP protocol. Every application using this sockets protocol can connect to the GMV GNSS module.

The mentioned services and their involved algorithms and structures provided by the TOE are presented below:

TOE service	Algorithm	OSNMA service involved
Message Digest	SHA256, Merkle Tree	Public key authentication using Merkle Tree structure with SHA256 hash function
Signature Authentication	ECDSA P-256 ECDSA P-521, SHA256, SHA512, SHA3-256	Signature authentication using public key
Message Authentication	CMAC AES256, HMAC SHA256	Message Authentication Code verification

Table 4 OSNMA TOE Services

The GMV GNSS Module recollects and stores logs using a circular log file into the file system to give traceability of the invoked cryptographic services and the users involved.

3.4. NON-TOE OVERVIEW

The TOE provides an external API that **exclusively operates** via a TCP/IP socket connection for interfacing with the module. Therefore, any application with the capability to form a socket connection with the TOE is deemed a **client of the TOE.** The primary hardware and software requirements for such a client are **limited to the necessities for setting up this specific socket connection.**

3.5. TOE DELIVERING

The TOE is a comprehensive software suite provided by GMV. It is packaged and delivered as a **.zip** file named "**gmv-gnss-cryptographic-module-v2.1.7.zip**". Distribution occurs via encrypted email, necessitating the exchange of GPG public keys between the vendor and client beforehand.

3.5.1. TOE PHYSICAL SCOPE

The TOE, being a software package, is encapsulated within the contents of the provided .zip file. The TOE comprises three essential components: **cryptomodule_1.elf, cryptomodule_2.bin**, and **cryptomodule_3.bin**, along with a comprehensive set of documentation as outlined in RD.22 - v2.1.7. Upon receipt, a client will obtain a release file including these components compressed into a .zip format, denoted as ***gmv-gnss-cryptographic-module-v2.1.7.zip***, containing:

[Folder] GMV_GNSS_Cryptographic_Module_v2.1.7: The release TOE file that includes **all the constituent files of the TOE:**

[File] cryptomodule_1.elf

[File] cryptomodule_2.bin

[File] cryptomodule_3.bin

[Folder] documentation: Folder that includes all the documentation specified in RD.22 -v2.1.7

[File] GMV GNSS Cryptographic Module – User Guide.pdf: A comprehensive guide detailing file functionalities and module usage.

[File] GMV GNSS Cryptographic Module – Functional Spetification.pdf: A comprehensive guide detailing the TOE functionalities, how to build the module packets and more interesting information.

[File] hashes_file.txt: This file contains the unique hashes for each delivered file, enabling clients to verify their integrity. The hashes included in this file are:

File	Version	Format	SHA-256 Hash
GMV_GNSS_Cryptographic_Module_v2.1.7/ cryptomodule_1.elf	v2.1.7	.elf	8d67bb675277e365296026dc50169b917bf1a80209c7398f03d4c26ba55fb999
GMV_GNSS_Cryptographic_Module_v2.1.7/ cryptomodule_2.bin	v2.1.7	.bin	5095f19589331385d8075f1a9f9e9db49d380a77ee7b6fb483d324af0d4cf419
GMV_GNSS_Cryptographic_Module_v2.1.7/ cryptomodule_3.bin	v2.1.7	.bin	30efedec1fa03b33a00458ac1b263147389e01d2646063a5305c3f86410be27e
GMV_GNSS_Cryptographic_Module_v2.1.7.zip	v2.1.7	.zip	Indicated in hashes_file.txt, because it is cyclical and cannot be indicated in this document.

GMV GNSS Cryptographic Module – User Guide.pdf	v2.1.7	.pdf	343db2e3d197dfc071d815ece355ea284a171abf5cae2c5cee341a7cc141b762
GMV GNSS Cryptographic Module – Functional Specification.pdf:	v2.1.7	.pdf	1b2a3749606a055f375f4530df7eca1be12bfea4a864af31d9c2ec2d9348cf90

Table 5 Hash Values for Client Release Files

3.5.2. TOE LOGICAL DESCRIPTION

The TOE is composed of the following software components:

Crypto library: Component responsible for providing the cryptographic functions and the security services for the TOE (See Table 10). The services offered include the following:

- Memory Manager: This module is designed for the secure management of all data stored and utilized within the system. Key responsibilities encompass the zeroization of data, ensuring that sensitive information is effectively erased and rendered unrecoverable.
- State Controller: Implements the logic to control the state changes of the TOE and the authorized functions for each state. The *State Changer* function checks that only state changes authorized by the state machine logic is performed. The state logic is specified in the diagram below.

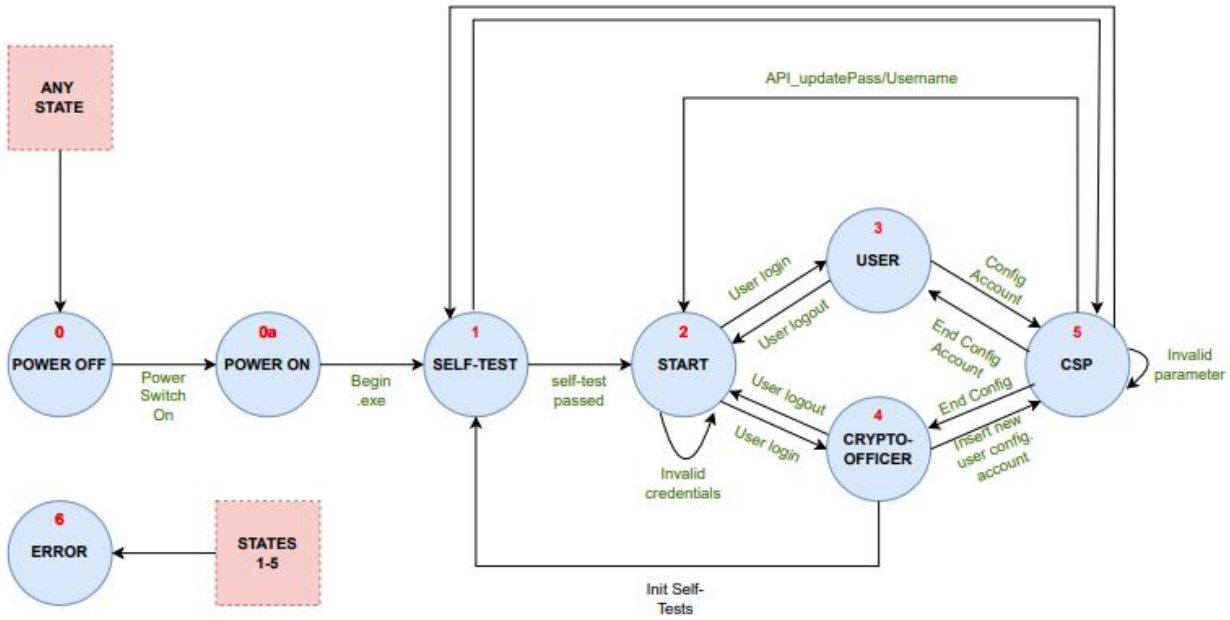

Figure 1 TOE SFM

- Self-tests: Tests the integrity of the executable code and the correct operation of the cryptographic functions. For integrity verification, the module computes a hash of the entire executable code and retains it. To validate the effectiveness of the cryptographic functions, it implements FIPS test vectors from the Cryptographic Algorithm Validation Tests (CAVS). These tests are conducted both prior to the module providing services to a client and whenever an authorized user initiates a "self-test operation."

- Timer: The main function of this module is to prevent Denial of Service (DoS) attacks. It achieves this by monitoring socket connections and automatically closing them if a specified time elapses without any operation requests.
- File system manager: This module is responsible for manager the persistence of the system. Its primary function involves ensuring that Role-Based Access Control (RBAC) is applied to all operations, including the addition, modification, and deletion of stored data.
- Handler: Facilitates the management of all internal requests between the module and the crypto library, ensuring the maintenance of the secure channel.

User Management Module: This module is responsible for user authentication and the comprehensive management of user profiles and their associated data, including usernames, passwords, and roles.

All the components that are included in the scope of this evaluation are in red in the following figure.

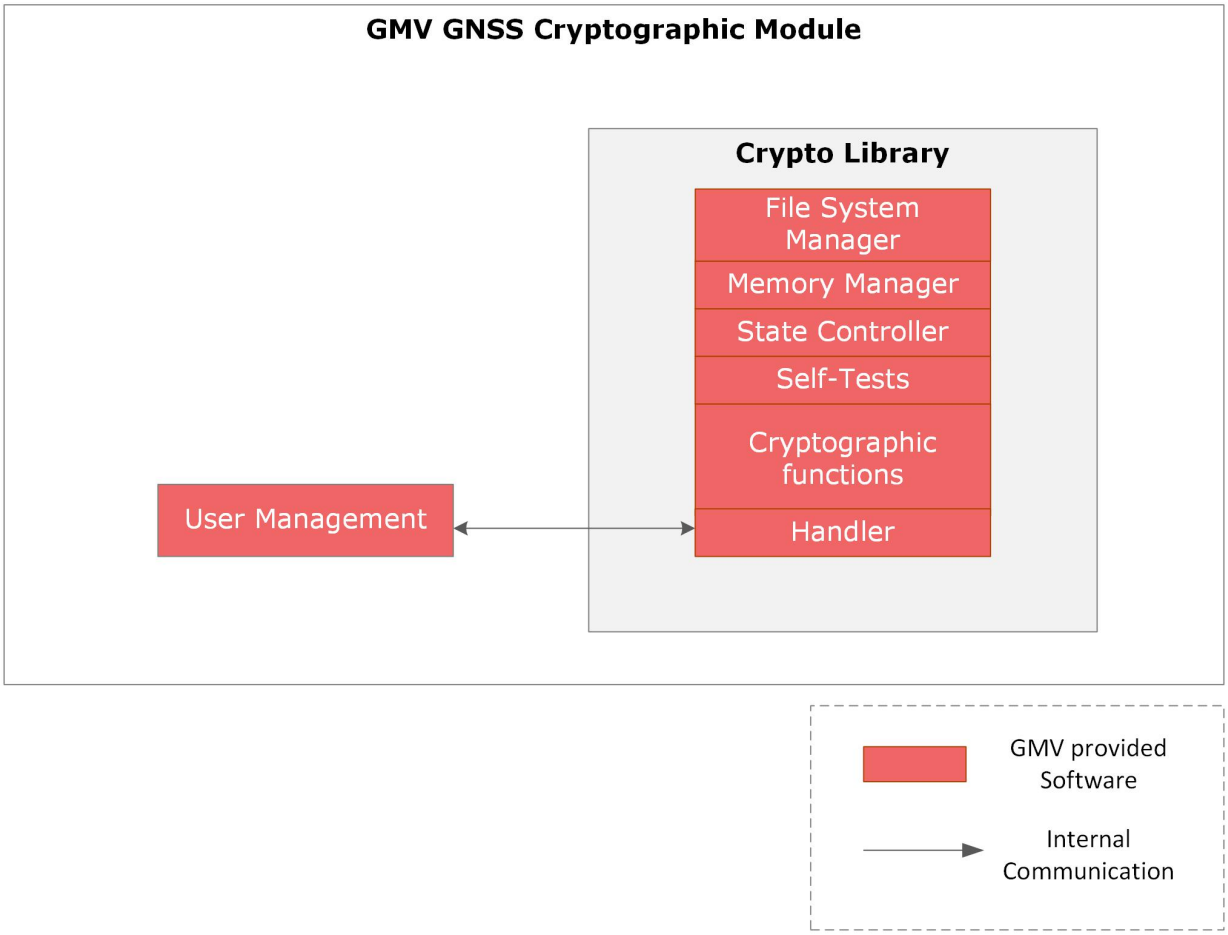

Figure 2 TOE Logical Architecture

4. CONFORMANCE CLAIMS

4.1. CC CONFORMANCE CLAIM

This security target claims conformance to the Common Criteria 2017, version 3.1, revision 5, with:

CC 2017 Part 1 [RD.1 - v2.1.7]

CC 2017 Part 2 [RD.2 - v2.1.7]

CC 2017 Part 3 [RD.3 - v2.1.7]

4.2. PP CLAIM

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

4.3. PACKAGE CLAIM

This Security Target claims conformance to the assurance package EAL2 specified in CC 2017 Part 3. [RD.3 - v2.1.7]

4.4. CONFORMANCE RATIONALE

No conformance rationale is necessary for this evaluation since this Security Target does not claim conformance to any Protection Profile.

5. SECURITY PROBLEM DEFINITION

This section identifies assets as **R.*asset*** assumptions as **A.*assumption***, threat agents as **TA.*agent*** , threats as **T.*threat*** and policies as **P.*policy***.

5.1. ASSETS TO BE PROTECTED

The TOE shall protect the following assets:

R.USER_DATA: Data supplied by a client for use in the TOE services. This data requires confidentiality and integrity, in case of users' passwords and integrity in case of the other associated data.

R.OSNMA_KEYS: Public keys and TESLA keys used by the TOE in support of the cryptographic services. The integrity of these keys must be protected.

R.AES_KEY: Secret key used to encrypt and decrypt the communication path between the TOE and the user, it must be protected in integrity and confidentiality.

R.INTEGRITY_HASH: Imported hash to verify the integrity of the embedded code, this data requires integrity.

R.SOFTWARE: Embedded software of the GMV GNSS Module. The software implements the security mechanisms of the TOE; therefore it must be protected in integrity.

5.2. THREATS

5.2.1. THREAT AGENTS

The following threat agents have been identified for the security of the TOE:

TA.INSIDER: Privileged user which intention is to compromise the TOE assets and/or security mechanisms (e.g. Crypto Officers).

TA.USER: Authorized user that commits errors or are absent-mindedness that affects the TOE security (e.g. Crypto Officers, Users).

TA.EXTERNAL: Unauthorized user that interacts with the TOE through the remote channel intending to corrupt the security mechanisms of the TOE (e.g. Cybercriminals). This thread can be mitigated with a correct module access control.

5.2.2. THREATS DESCRIPTION

5.2.2.1. THREATS ON CRITICAL SECURITY PARAMETERS

T.KEY_ALTERATION **Alteration of keys**	TA.EXTERNAL or TA.INSIDER might modify or alter all or part of R.OSNMA_KEYS, R.INTEGRITY_HASH and R.AES_KEY by interaction with the TOE logical functions, the file system in which this information is stored or

within the TOE environment. TA.USER with Crypto-officer role might also modify or alter R.AES_KEY since this user has the responsibility to load this data in the GMV GNSS Module.

T.KEY_DISCLOSE
Disclosure of keys

TA.EXTERNAL or TA.INSIDER might disclose all or part of R.OSNMA_KEYS, R.AES_KEY by means of intercept the data in the network communication channel, reading the information in the file system or even during the use of the TOE services.

5.2.2.2. DATA THREATS

T.DATA_MANIPUL
Manipulating data

TA.EXTERNAL , TA.INSIDER or TA.USER could potentially tamper with R.USER_DATA or R.OSNMA_KEYS during their transfer from the client application to the TOE (in the case of TA.EXTERNAL) or within the TOE's persistence system (in the case of TA.INSIDER and TA.USER). Such manipulation might adversely impact the performance of cryptographic functions, as the TOE may not be able to detect the presence of corrupted data.

T.MEMORY_DUMP_ATTACK
Deallocated memory access

TA.EXTERNAL , TA.INSIDER or TA.USER might access to the module's deallocated memory. This may allow access to previous security information such as R.USER_DATA, R.OSNMA_KEYS or R.AES_KEY.

5.2.2.3. SERVICE THREATS

T.BLOCK_SERVICE
Blocking of TOE services

TA.EXTERNAL , TA.INSIDER or TA.USER might disable the TOE service through open a socket connection and not operating with the module, or requesting infinite operations . This may hold the communication channel and block another client to connect with the TOE.

T.MALICIOUS_SW
Malicious software

TA.EXTERNAL or TA.INSIDER might modify or alter the R.SOFTWARE by loading malicious software by means of the interaction with the TOE logical interfaces or getting permissions of the operative system in which the TOE is operating.

5.3. ORGANIZATIONAL SECURITY POLICIES

The TOE meets the following organizational security policies:

P.ALGORITHMS
Approved cryptographic

The TOE provides cryptographic functions endorsed by ESA authority as suitable for user implementation. Detailed descriptions of these algorithms

algorithms	can be found in Section 2.2 of RD.13-v2.1.7.
P.AUDIT **Audit trail generation**	The TOE is required to generate an audit trail of security-relevant events, exporting the events details and the subject associated with the event.

5.4. ASSUMPTIONS

This section describes the security aspects of the environment in which the TOE is intended to be used. The TOE is assured to provide effective security measures in a co-operative non-hostile environment only if it is installed, managed, and use correctly. The following specific conditions are assumed to exist in an environment where TOE is employed:

A.ADMINISTRATOR **Reliable administrator**	The administrator of TOE will be reliable and will operate the TOE in a secure and correct manner, avoiding any harmful intent to the device or its data.
A.ENVIRONMENT **Environment protection**	The TOE operates in a protected environment that limits physical access to the TOE to authorised Administrators. The TOE environment ensures the confidentiality, integrity, and availability of the security relevant data for the TOE initialisation, start-up, and operation. (e.g. the verification data that allows check the integrity of the TOE software)
A.OPERATING_SYSTEM **Operating System**	The operating system has been selected to provide the functions required by the TOE and the security mechanisms necessary to assure the integrity of TOE code.
A.TIMESTAMP **Operating System Timestamp**	The TOE runs on a platform that provides reliable timestamps.
A.LOG_TRAIL **Availability of log records**	The log records will be accessible to the TOE administrator, who can directly retrieve them from the TOE's persistence system.

6. SECURITY OBJECTIVES

This section identifies and defines the security objectives for the TOE and its operational environment. These Security Objectives reflect the intention to reduce the risks associated to the threats specified above, as well as comply with the organisational security policies and assumptions.

6.1. SECURITY OBJECTIVES FOR THE TOE

The IT security objectives for the TOE are addressed below:

OT.ALGORITHMS **Use of approved cryptographic algorithms**	The TOE offers cryptographic function provided for users that are validated by the ESA authority as appropriate.
OT.AUDIT **Generation and storage Audit data**	The TOE shall audit the following events: - TOE initialization and shutdown - Module state changes - Self-test operation - User login/logout - Cryptographic operations - Addition, deletion, and modifications of the security attributes and data stored in the TOE. - Zeroization - Memory violation - Establishment and disconnection of the secure channel
OT.DOS_RESPONSE **Denial of Service attack response**	The TOE shall be able to identify a service interruption attack, and act so the service can be established and operate in the common manner.
OT.RESTRICT_ACCESS **Control access to TOE services**	The TOE shall restrict the access to its services, depending on the user role, to those services explicitly assigned to this role. Also, it shall restrict access when a specific timeout is over.
OT.SEC_CHANNEL **Exchange of encrypted data**	Confidentiality and integrity of the data transferred between the TOE and the user are ensured by the secure communication channel. The TOE shall encrypt the data sent through the channel, and decrypt the data received by the channel.

OT.SEC_DATA **Security of data**	The confidentiality and integrity of the R.USER_DATA, R.AES_KEY and R.INTEGRITY_HASH, in addition to the integrity of the R.OSNMA_KEYS shall be ensured during their whole lifetime.
OT.SEC_STATE **Secure State in case Error is detected**	The TOE shall enter a secure state whenever it detects a failure or an integrity error of software. The secure state shall prevent the loss of confidentiality and integrity of R.USER_DATA, R.AES_KEY and R.INTEGRITY_HASH, and the integrity of R.OSNMA_KEYS
OT.USER_AUTH **Authentication of users interacting with the TOE**	The TOE shall be able to identify and authenticate the users acting with a defined role, before allowing access to any TOE service. Identification shall be password-based, and authentication shall be role-based.

6.2. SECURITY OBJECTIVES FOR THE OPERATIONAL ENVIRONMENT

The security objectives for the operational environment are addressed below:

OE.APPLICATION **Security in Client Application**	The applications which use the TOE shall perform the necessary security checks on the data passes through the secure channel to the TOE. They shall ensure the appropriate format for the communication and operations.
OE.AUDIT_REVIEW **Audit review**	The environment ensures the availability of the generated by the TOE audit trails and provides a review of the audit trail recorded by the TOE.
OE.OPERATING_SYSTEM **Reliable Operating System**	The Operating system in which the TOE is allocated shall enable the correct operation of the TOE services and establish the necessary mechanism to ensure the security of its embedded code. It shall also provide reliable timestamps for the use of the TOE.
OE.PERSONNEL **Reliable Personnel**	The personnel using the TOE shall be aware of the obligations they have to face, depending on their role. The personnel shall be trained on correct usage of the TOE and should be trusted people.
OE.PROTECT_ACCESS **Prevention of unauthorised physical access**	The TOE shall be protected by physical, logical, and organisational protection measures, to prevent any TOE modification, as well as protect assets disclosure. Those measures shall restrict the TOE usage to authorised persons only.

6.3. SECURITY OBJECTIVES RATIONALE

In this section, we provide a comprehensive rationale for the established security objectives within the context of the TOE (Target of Evaluation) and its operational environment.

6.3.1. SECURITY OBJECTIVES FOR THE TOE RATIONALE

OT.ALGORITHMS

- **P.ALGORITHMS:** This objective directly aligns with Policy P.ALGORITHMS, as it mandates the use of approved cryptographic algorithms. By ensuring the TOE uses approved algorithms, it mitigates threats related to cryptographic weaknesses.

OT.AUDIT

- **P.AUDIT:** This objective directly aligns with Policy P.AUDIT, as it mandates the generation of audit trails, monitoring and recording security events.

OT.DOS_RESPONSE

- **T.BLOCK_SERVICE:** This objective that the TOE can respond to service interruption attacks, aligning with the broader security goal of maintaining service availability.

OT.RESTRICT_ACCESS

- **T.MALICIOUS_SW:** By controlling access to TOE services based on user roles and enforcing proper authorization mechanisms, the TOE can minimize the risk of unauthorized users introducing malicious software into the system. Restricting access ensures that only trusted and authorized individuals can interact with the TOE, reducing the likelihood of software tampering.

- **T.KEY_DISCLOSE:** By controlling access to TOE services, the TOE can limit the exposure of cryptographic keys and sensitive data to only authorized users, in this case, users with crypto officer role.

- **T.KEY_ALTERATION:** Unauthorized access to TOE services can lead to the modification or alteration of cryptographic keys, potentially compromising data integrity and security. By enforcing access restrictions, the TOE can limit the opportunities for individuals, including insiders, to tamper with these keys, ensuring their integrity and confidentiality.

- **T.DATA_MANIPUL:** By controlling access to TOE services based on user roles and implementing authorization mechanisms, the TOE can prevent unauthorized data manipulation, ensuring the integrity and security of sensitive information.

- **T.BLOCK_SERVICE:** By enforcing access restrictions when a timeout is reached, this objective prevents service interruption attacks that that attempt to take control of the system, thereby denying access to other users.

OT.SEC_CHANNEL

- **T.MALICIOUS_SW:** Establishing a secure channel effectively reduces the risk of unauthorized users introducing malicious software into the system.

- **T.DATA_MANIPUL:** Establishing a secure channel helps prevent unauthorized users from gaining access to sensitive information during data transmission.

OT.SEC_DATA

- **T.KEY_DISCLOSE:** By focusing on the security of data, this objective helps protect cryptographic keys from disclosure threats, ensuring the confidentiality and integrity of the keys and the data they protect.

- **T.KEY_ALTERATION**: By maintaining the security of data, these objective safeguards cryptographic keys from alterations, thereby preserving their integrity and ensuring the security of data encrypted with those keys.

- **T.DATA_MANIPUL:** This security objective ensures that data is securely stored and that any manipulation attempts, whether by external threat agents or insiders, are thwarted.

OT.SEC_STATE

- **T. MALICIOUS_SW:** By restricting the operational states in which the TOE can operate, this objective effectively reduces the potential for malicious software to be introduced into the TOE.

- **T.DATA_MANIPUL:** By limiting the operational states in which the TOE can manipulate data, this objective can effectively minimize the risk of unintended data manipulation.

- **T. MEMORY_DUMPT_ATTACK:** If an attacker or an internal user (gains access to the module's deallocated memory, they may gain access to previously stored security information. In response to this threat, this objective strives to ensure that even in the presence of unauthorized access to deallocated memory, the TOE enters a secure state that safeguards the confidentiality and integrity of critical data.

OT.USER_AUTH

- **T.KEY_DISCLOSE:** By authenticating users before granting access to TOE services, it helps prevent unauthorized access to cryptographic keys or other sensitive data. Through user authentication, the TOE ensures that only authorized individuals can access critical resources, reducing the risk of key disclosure.

- **T.KEY_ALTERATION**: By authenticating users, the TOE ensures that only authorized individuals have access to cryptographic keys.

- **T.DATA_MANIPUL:** By authenticating users before allowing them to interact with the TOE, it ensures that only authorized users have access to data.

6.3.2. SECURITY OBJECTIVES FOR THE OPERATIONAL ENVIRONMENT

OE.APPLICATION

- **A.ADMINISTRATOR:** This objective relates with the assumption since it assumes that client application perform security checks to protect the TOE.

OE.AUDIT_REVIEW

- **A.LOG_TRAIL:** This objective relates with the assumption of the availability and review of audit trails.

OE.OPERATING_SYSTEM

- **A.OPERATING_SYSTEM:** This objective relates with the assumption of a reliable operating system in which the TOE will be executed.

- **A.TIMESTAMP:** This objective relates with the assumption of reliable timestamps provided by the operating system.

OE.PERSONNEL

- **A.ADMINISTRATOR:** This objective is linked to Assumption A.ADMINISTRATOR, assuming trustworthy personnel.

- **T.KEY_DISCLOSE:** Personnel who are reliable and trained are less likely to engage in actions that could lead to key disclosure, such as sharing sensitive information with unauthorized parties.

- **T.DATA_MANIPUL:** Trustworthy and trained personnel are more likely to follow secure practices, reducing the risk of data manipulation.

OE.PROTECT_ACCESS

- **A.ENVIRONMENT:** This objective aligns with the assumption of protection measures to prevent unauthorized physical access.

	OT.ALGORITHMS	OT.AUDIT	OT.DOS_RESPONSE	OT.RESTRICT_ACCESS	OT.SEC_CHANNEL	OT.SEC_DATA	OT.SEC_STATE	OT.USER_AUTH	OE.APPLICATION	OE.AUDIT_REVIEW	OE.OPERATING_SYSTEM	OE.PERSONNEL	OE.PROTECT_ACCESS
T.MALICIOUS_SW				X	X		X						
T.KEY_DISCLOSE				X		X		X				X	
T.KEY_ALTERATION				X		X		X					
T.MEMORY_DUMP_ATTACK							X						
T.DATA_MANIPUL				X	X	X	X	X				X	
T.BLOCK_SERVICE			X	X									
P.ALGORITHMS	X												
P.AUDIT		X											
A.ADMINISTRATOR									X			X	
A.ENVIRONMENT													X
A.OPERATING_SYSTEM											X		
A.TIMESTAMP											X		
A.LOG_TRAIL										X			

Table 6 Security objectives rationale

7. SECURITY REQUIREMENTS

This section outlines the security requirements for the TOE and its operational environment. The requirement presented here are derived from the Common Criteria 2017, version 3.1, revision 5, Part 2 [RD.2 - v2.1.7]. While the content is reproduced here verbatim, adjustments have been made to the identifiers to ensure consistency with the iteration conventions used in this document. Specific naming conventions have been implemented as necessary to distinctly indicate various operations.

Conventions

Assignment: Assignments are identified in brackets and bold (e.g., **[assigned value]**).

Selection: Selections are identified in brackets, bold, and italics (e.g., ***[selected value]***). Assignments within selections are identified using the previous conventions, except that the assigned value would also be italicized, and extra brackets would occur (e.g., ***[selected value [assigned value]]***).

Refinement: Refinements are identified using bold text (e.g., **added text**) for additions and strike-through text (e.g., ~~deleted text~~) for deletions.

7.1. TOE SECURITY FUNCTIONAL REQUIREMENTS

The Security Functional Requirements included in this section are derived from Part 2 of the Common Criteria [RD.2 - v2.1.7]. The requirements applicable to the TOE are the following:

Class name	Class family	Description
SECURITY AUDIT (FAU)	FAU_GEN	Audit data generation
	FAU_STG	Audit event storage
CRYPTOGRAPHIC (FCS)	FCS_COP	Cryptographic operation
	FCS_CKM	Cryptographic key management
USER DATA PROTECTION (FDP)	FDP_ITC	Import from outside of the TOE
	FDP_RIP	Residual information protection
	FDP_UCT	Inter-TSF user data confidentiality transfer protection
	FDP_ACC	Access control policy
	FDP_ACF	Access control functions
IDENTIFICATION AND AUTHENTICATION (FIA)	FIA_AFL	Authentication failures
	FIA_ATD	User attribute definition
	FIA_SOS	Specification of secrets
	FIA_UAU	User authentication
	FIA_UID	User identification
SECURITY MANAGEMENT (FMT)	FMT_MSA	Management of security attributes
	FMT_SMF	Specification of management functions
	FMT_SMR	Security management roles
PROTECTION OF THE TSF (FPT)	FTP_TST	TSF self-test

RESOURCE UTILISATION (FRU)	FRU_RSA	Resource allocation
TOE ACCESS (FTA)	FTA_SSL	Session locking and termination
	FTA_TSE	TOE session establishment

Table 7 Security Functional Requirements

7.1.1. SECURITY AUDIT (FAU)

7.1.1.1. 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 shutdown of the audit functions.

b) All auditable events for ***[basic]*** level of audit; and

c) [All auditable events described in Table 8 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 (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 ST record **[additional audit described in Table 8 Auditable events]**

Requirement	Auditable Events	Additional audit record
FCS_CKM.4	Any attempt to execute zeroization function.	Subject attributes (e.g. username).
FCS_COP.1	All use of cryptographic operations	Any applicable cryptographic mode(s) of operation and subject attributes (e.g. username).
FDP_ITC.2	Any attempts to import data.	All attempts to import user data, including any security attributes (e.g. username)
FIA_AFL.1	Any attempt that exceeds the unsuccessful authentication try.	Subject attributes (e.g. username).
FIA_SOS.1	Failure creating or changing a password.	Subject attributes (e.g. username).
FIA_UAU.1	Any attempts to login the TOE.	Subject attributes (e.g. username). In case of failure, description of the error.
FIA_UID.1	Any attempts to login the TOE.	Subject attributes (e.g. username). In case of failure, description of the error.
FMT_MSA.1	All modifications of user data and the security attributes	Subject attributes (e.g. username). And object attributes (e.g. username). In case of failure, description of the error.
FMT_MSA.3	All attempt to modify the values of security attributes	Subject attributes (e.g. username) And object attributes (e.g. username). In case of failure, description of the error.
FPT_TST.1	Execution of the TSF self-tests and the results of the tests	Subject attributes (e.g. username). In case of failure, description of the error.
FRU_RSA.1	Any attempt to exceed the maximum memory	Subject attributes (e.g. username). In case of

	quota. Any attempt to execute zeroization function.	failure, description of the error.
FTA_SSL.3	Termination of an interactive session by the session locking mechanism.	Subject attributes (e.g. username).
FTA.SSL.4	Termination of an interactive session by the user	Subject attributes (e.g. username).
FTA_TSE.1	All attempts at establishment of a user session	Subject attributes (e.g. username). In case of failure, description of the error.

Table 8 Auditable events

7.1.1.2. FAU_STG.1 PROTECTED AUDIT TRAIL STORAGE

FAU_STG.1.1 The TSF shall protect the stored audit records in the audit trail from unauthorised deletion.

FAU_STG.1.2 The TSF shall be able to ***[prevent]*** unauthorised modifications to the stored audit records in the audit trail.

7.1.2. CRYPTOGRAPHIC KEY MANAGEMENT (FCS_CKM)

7.1.2.1. 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 **[zeroization]** that meets the following: **[None].**

7.1.3. CRYPTOGRAPHIC OPERATION (FCS_COP)

7.1.3.1. FCS_COP.1 CRYPTOGRAPHIC OPERATION

FCS_COP.1.1 The TSF shall perform **[Column Cryptographic operations from Table 9 Cryptographic algorithms and operations]** in accordance with a specified cryptographic algorithm **[Column Cryptographic algorithm from Table 9 Cryptographic algorithms and operations]** and cryptographic key sizes **[Column Key size from Table 9 Cryptographic algorithms and operations]** that meet the following: **[Column Standards from Table 9 Cryptographic algorithms and operations]**.

Cryptographic algorithm	Key size / Hash size	Standards	Cryptographic operations
SHA	256, 512, 3-256	ISO/IEC 10118-3	Hash function and Merkle Tree verification protocol with 256 key-size
HMAC-SHA	256	ISO/IEC 9797-2	Authentication
ECDSA	256, 521	ISO/IEC 29167-16	Signature verification and Merkle Tree verification protocol with 256 key-size
CMAC-AES	256	RFC 4493	Verification
TESLA	-	RFC 4082	Authentication

ALGORITHM			

Table 9 Cryptographic algorithms and operations

7.1.4. USER DATA PROTECTION (FDP)

7.1.4.1. FDP_ITC.2 IMPORT OF USER DATA WITH SECURITY ATTRIBUTES

FDP_ITC.2.1 The TSF shall enforce the **[Role-Based Access Control (RBA) with login authentication as first interaction with the module]** when importing user data, controlled under the SFP, from outside of the TOE.

FDP_ITC.2.2 The TSF shall use the security attributes associated with the imported user data.

FDP_ITC.2.3 The TSF shall ensure that the protocol used provides for the unambiguous association between the security attributes and the user data received.

FDP_ITC.2.4 The TSF shall ensure that interpretation of the security attributes of the imported user data is as intended by the source of the user data.

FDP_ITC.2.5 The TSF shall enforce the following rules when importing user data controlled under the SFP from outside the TOE: **[**

User passwords must be encrypted using the SHA-256 algorithm by the TOE's corresponding TSF before storage.

Access logs for imported user data must be generate and store for audit purposes.

The TOE must enter the CSP_STATE before storing the imported data].

7.1.4.2. FDP_RIP.1 SUBSET RESIDUAL INFORMATION PROTECTION

FDP_RIP.1.1 The TSF shall ensure that any previous information content of a resource is made unavailable upon the ***[deallocation of the resource from]*** the following objects:

- **[Program variables that are no longer used.**
- **Any data deleted from the file system.]**

7.1.4.3. FDP_UCT.1 INTER-TSF USER DATA CONFIDENTIALITY TRANSFER PROTECTION

FDP_UCT.1.1 The TSF shall enforce the **[Role-Based Access Control (RBA) with login authentication as first interaction with the module**] to ***[transmit and receive]*** user data in a manner protected from unauthorized disclosure.

7.1.4.4. FDP_ACC.1 SUBSET ACCESS CONTROL

FDP_ACC.1.1 The TSF shall enforce the **[Role-Based Access Control (RBA)]** on **[**

Subjects: All users

Objects: R.OSNMA_KEYS and R.USER_DATA

Operations: Column Operations from Table 10]

7.1.4.5. FDP_ACF.1 SECURITY ATTRIBUTE BASED ACCESS CONTROL

FDP_ACF.1.1 The TSF shall enforce the **[role-based access control]** to objects based on the following: **[**

- Subjects: All users

- Subjects Attributes: Role.

- Objects: Content within File System files, cryptographic functions, and audit logs.

-Operations: Column Authorized role from Table 10.]

Operation	Authorized role
Modify user security attributes: - Role	Crypto Officer (CO)
Add, modify, or delete user attributes: -TESLA root key -public key	Crypto Officer (CO) User (Only its own associated attributes)
Perform Self-tests	Crypto Officer (CO)
Perform Cryptographic operations	Crypto Officer (CO) and User
Request the module state	Crypto Officer (CO) and User
Request module general information	Crypto Officer (CO) and User

Table 10 Authorized operations rationale.

FDP_ACF.1.2 The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: **[Role associated to the user authentication.]**

FDP_ACF.1.3 The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: **[No additional rules].**

FDP_ACF.1.4 The TSF shall explicitly deny access of subjects to objects based on the following additional rules: **[No additional rules].**

7.1.5. IDENTIFICATION AND AUTHENTICATION (FIA)

7.1.5.1. FIA_AFL.1 AUTHENTICATION FAILURE HANDLING

FIA_AFL.1.1 The TSF shall detect when **[[*three*]]** unsuccessful authentication attempts occur related to **[Crypto Officer or User authentication attempts].**

FIA_AFL.1.2 When the defined number of unsuccessful authentication attempts has been ***[met]***, the TSF shall **[audit the error and close the communication path].**

7.1.5.2. FIA_ATD.1 USER ATTRIBUTE DEFINITION

FIA_ATD.1.1 The TSF shall maintain the following list of security attributes belonging to individual users: **[role]**.

7.1.5.3. FIA_SOS.1 VERIFICATION OF SECRETS

FIA_SOS.1.1 The TSF shall provide a mechanism to verify that secrets meet: **[**

- **Must contain at least eight characters.**
- **Must contain at least one digit.**
- **Must contain at least one uppercase.**
- **Must contain at least one lowercase.**
- **Must contain at least one special character.**
- **Must not contain spaces.]**

7.1.5.4. FIA_UAU.1 TIMING OF AUTHENTICATION

FIA_UAU.1.1 The TSF shall allow **[establish the secure channel connection between user and the module]** on behalf of the user to be performed before the user is authenticated.

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

7.1.5.5. FIA_UID.1 TIMING OF IDENTIFICATION

FIA_UID.1.1 The TSF shall allow **[establish the secure channel connection between user and the module]** on behalf of the user to be performed before the user is identified.

FIA_UID.1.2 The TSF shall require each user to be successfully identified before allowing any TSF-mediated actions on behalf of that user.

7.1.6. SECURITY MANAGEMENT (FMT)

7.1.6.1. FMT_MSA.1 MANAGEMENT OF SECURITY ATTRIBUTES

FMT_MSA.1.1 The TSF shall enforce the **[Role-Based Access Control]** to restrict the ability to ***[modify]*** the security attributes **[role]** to **[Crypto Officer].**

7.1.6.2. FMT_MSA.3 STATIC ATTRIBUTE INITIALIZATION

FMT_MSA.3.1 The TSF shall enforce the **[Role-Based Access Control]** to provide ***[restrictive]*** default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2 The TSF shall allow the **[Crypto Officer]** to specify alternative initial values to override the default values when an object or information is created.

7.1.6.3. FMT_SMF.1 SPECIFICATION OF MANAGEMENT FUNCTIONS

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

- **Add, modify, or delete user attributes (username, password, or keys (TESLA root key and public key))**
- **Modify user security attributes (role)].**

7.1.6.4. FMT_SMR.1 SECURITY ROLES

FMT_SMR.1.1 The TSF shall maintain the roles: **[**

- **Crypto Officer: Role of administrator. Can perform all the functions specified in Table 10 Authorized operations .**
- **User: Role of normal user. It can only perform the functions specified in Table 10 Authorized operations .]**

FMT_SMR.1.2 The TSF shall be able to associate users with roles.

7.1.7. PROTECTION OF THE TSF (FPT)

7.1.7.1. FPT_TST.1 TSF SELF TEST

FPT_TST.1.1 The TSF shall run a suite of self-tests ***[during initial start-up, at the request of the authorized user]*** to demonstrate the correct operation of ***[the TSF]***: **[**

- **AES Test**
- **CMAC Test**
- **ECDSA Test**
- **HMAC Test**
- **SHA Test**
- **TESLA Test**

- **Integrity Test]**

FPT_TST.1.2 The TSF shall provide authorised users with the capability to verify the integrity of ***[[Cryptographic functions]]***

FPT_TST.1.3 The TSF shall provide authorised users with the capability to verify the integrity of ***[[Executable code]]***

7.1.7.2. FPT_TDC.1 INTER-TSF BASIC TSF DATA CONSISTENCY

FPT_TDC.1.1 The TSF shall provide the capability to consistently interpret **[**

Username

User's password

User's Root Key

User's Public Key] when shared between the TSF and another trusted IT product.

FPT_TDC.1.2 The TSF shall use **[**

SHA256 encryption to verify user's password during the authentication process.

An ID association mechanism to relate the user and their corresponding security attributes allocate within the TOE's file system] when interpreting the TSF data from another trusted IT product.

7.1.8. RESOURCE UTILISATION (FRU)

7.1.8.1. FRU_RSA.1 MAXIMUM QUOTAS

FRU_RSA.1.1 The TSF shall enforce maximum quotas of the following resources: **[packet size (30 Kilobytes), packet parameters size (1 Kilobyte) and timeout (10 minutes)]** that ***[individual user]*** can use ***[simultaneously].***

7.1.9. TOE ACCESS (FTA)

7.1.9.1. FTA_SSL.3 TSF-INITIATED TERMINATION

FTA_SSL.3.1 The TSF shall terminate an interactive session after **[10 minutes of inactivity].**

**It is considered inactivity the lack of request of operations from the user (lack of packets).*

7.1.9.2. FTA_SSL.4 USER-INITIATED TERMINATION

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

7.1.9.3. FTA_TSE.1 TOE SESSION STABLISHMENT

FTA_TSE.1.1 The TSF shall be able to deny session establishment based on **[an already active session or a user failure of authentication.]**

7.1.10. TRUSTED PATH (FTP_TRP)

7.1.10.1. FTP_TRP.1 TRUSTED PATH

FTP_TRP.1.1 The TSF shall provide a communication path between itself and ***[remote]*** users that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from ***[modification, disclosure].***

FTP_TRP.1.2 The TSF shall permit ***[remote users]*** to initiate communication via the trusted path.

FTP_TRP.1.3 The TSF shall require the use of the trusted path for ***[initial user authentication, [***

Execute User Management functions (Create/Delete User, Modify User's security attributes).

Execute Cryptographic functions]].

7.2. TOE SECURITY ASSURANCE REQUIREMENTS

The selected package of security assurance requirements is EAL2, described in CC Part 3 RD.3-v2.1.7 . The corresponding requirements are listed in table below.

Assurance Class	Assurance Components
ADV: Development	ADV_ARC.1 Security architecture description
	ADV_FSP.2 Security-enforcing functional specification
	ADV_TDS.1 Basic Design
AGD: Guidance documents	AGD_OPE.1 Operational user guidance
	AGD_PRE.1 Preparative procedures
ALC: Life-cycle support	ALC_CMC.2 Use of a CM system
	ALC_CMS.2 Parts of the TOE CM coverage
	ALC_DEL.1 Delivery procedures
ASE: Security Target evaluation	ASE_CCL.1 Conformance claims
	ASE_ECD.1 Extended components definition
	ASE_INT.1 ST introduction
	ASE_OBJ.2 Security objectives
	ASE_REQ.2 Derived security requirements
	ASE_SPD.1 Security problem definition
	ASE_TSS.1 TOE summary specification

ATE: Tests	ATE_COV.1 Evidence of coverage
	ATE_FUN.1 Functional testing
	ATE_IND.2 Independent testing - sample
AVA: Vulnerability assessment	AVA_VAN.2 Vulnerability analysis

Table 11 Assurance requirements

7.3. SECURITY REQUIREMENTS RATIONALE

7.3.1. SECURITY FUNCTIONAL REQUIREMENTS

This section detailed rationales between SFR and the security objectives described in section 6.

OT.ALGORITHMS

- **FCS_CKM.4** ensures that the deletion of cryptographic keys is securely executed in compliance with specified security policies, as indicated in OT.ALGORITHMS.
- **FCS_COP.1** ensures that cryptographic operations are carried out securely and meet specified security policies as mentioned in OT.ALGORITHMS.

OT.AUDIT

- **FAU_GEN.1** corresponds to the generation of audit data as required by the TOE in OT.AUDIT, including events such as initialization, module state changes, user login/logout, and more.
- **FMT_SMF.1** is linked to security management functions. Those related to generation of log are directly aligned with the OT_AUDIT objective.
- **FAU_STG.1** corresponds to the storage of audit data as required by the TOE in OT.AUDIT.

OT.DOS_RESPONSE

- **FTA_SSL.3** deals with protection against denial of service (DoS) attacks, as inactive session, which aligns with the requirement for the TOE to identify and respond to service interruption attacks as specified in OT.DOS_RESPONSE.

OT.RESTRICT_ACCESS

- **FDP_ITC.2** is associated with user data importation. All the importation process must be executed by the authorized roles, as it is specified in OT.RESTRICT_ACCESS
- **FDP_ACC.1** is associated with access control policies, which align with the requirement to restrict access based on user roles as specified in OT.RESTRICT_ACCESS.
- **FDP_ACF.1** relates to access control functions, which align with the requirement to restrict access based on user roles as specified in OT.RESTRICT_ACCESS.

- **FIA_AFL.1** relates to deny access after authentication failure, which is related to the control access to TOE services specified in OT.RESTRICT_ACCESS

- **FIA_ATD.1** is closely related to user attribute definitions. In the context of OT.RESTRICT_ACCESS, the role-based access control implies that individual user security attributes will correspond to their respective roles.

- **FMT_MSA.1** is directly connected to the management of security attributes, with access limited to authorized users based on their assigned roles, as specified in OT.RESTRICT_ACCESS.

- **FMT_MSA.3** is related to the initialization of security attributes management. In accordance with the principles outlined in OT.RESTRICT_ACCESS, the modification of these security attributes is restricted solely to authorized users. This ensures that only authorized personnel have the capability to alter these critical security attributes..

- **FMT_SMF.1** is linked to security management functions. Those related to the creation, modification, or deletion of security attributes are directly aligned with the access control policies specified in OT.RESTRICT_ACCESS.

- **FMT_SMR.1** relates to security roles, which align with the requirement to restrict access based on user roles as specified in OT.RESTRICT_ACCESS.

- **FPT_TST.1** is intricately linked to testing procedures and the initiation of a secure state in response to errors or failures. This correlation aligns with the requirement that only authorized users have the capability to request integrity validation.

OT.SEC_CHANNEL

- **FDP_ITC.2** is associated with user data importation. As it is required in OT_SEC_CHANNEL all the information to be imported must be encrypted in the communication channel.

- **FIA_UAU.1** corresponds to the establishment of a secure communication channels before the user authentication, which is related with the requirement specified in OT.SEC_CHANNEL.

- **FDP_UCT.1** is directly relevant to the OT.SEC_CHANNEL requirement as it addresses the imperative need to maintain data confidentiality during the transmission process.

- **FIA_UID.1** is related to the user identification process before allowing any action. The OT.SEC_CHANNEL implies the establishment of a secure channel to facilitate the user identification process.

- **FRU_RSA.1** relates to establish maximum quotas for resources. As it is outlined in OT.SEC_CHANNEL, all the data transmitted by the secure channel must be in the appropriate format.

- **FPT_TDC.1** relates to the capability to interpret the data shared through the trusted channel. As OT.SEC_CHANNEL established the necessity for secure communication channels, FPT_TDC.1 ensures that the TSF can consistently interpret and process the specific user data shared through this secure channel.

- **FTP_TRP.1** relates to establish a trusted path between remote users and the TOE, what is directly related with OT.SEC_CHANNEL.

OT.SEC_DATA

- **FMT_MSA.3** pertains to the management of security attributes initialization. These security attributes must be protected during their whole lifetime, as it is specified in OT.SEC_DATA.

OT.SEC_STATE

- **FDP_ITC.2** is associated with user data importation. All the importation process must be executed in CSP_STATE, therefore this SFR is related with the security objective OT.SEC_STATE

- **FPT_TST.1** is closely tied to the testing procedures and the initiation of a secure state in response to errors or failures. This aligns with the requirement stated in OT.SEC_STATE, which mandates the performance of self-tests following system initialization or authorized user requests, along with the initiation of a secure state upon the detection of specific errors.

- **FDP_RIP.1** focuses on prohibiting internal data flows within the TOE, which is essential to fulfil the objectives outlined in OT.SEC_STATE.

- **FTA_SSL.4** specifically addresses user-initiated termination of sessions. This aligns with the state machine architecture of the TOE, which incorporates distinct states for user log-in and log-out processes.

- **FTA_TSE.1** is directly associated with denying session establishment based on authentication process failures. The state machine architecture of the TOE is designed to enable the denial of session establishment if the system is not in the appropriate state.

OT.USER_AUTH

- **FIA_SOS.1** is focused on ensuring the security attributes of the system. Given that user identification relies on password-based authentication, as mentioned in OT.USER_AUTH, it becomes crucial to emphasize that user passwords must be strong enough to uphold the overall security of the system.

- **FIA_UAU.1** is associated with user authentication before any action which corresponds with the requirement described in OT.USER_AUTH.

- **FIA_UID.1** is associated with the user identification process before granting any actions. As specified in OT.USER_AUTH, this identification process is explicitly based on password authentication.

The following table is a summary of the mapping of the SFRs with the security objectives for the TOE.

	OT.ALGORITHMS	OT.AUDIT	OT.DOS_RESPONSE	OT.RESTRICT_ACCESS	OT.SEC_CHANNEL	OT.SEC_DATA	OT.SEC_STATE	OT.USER_AUTH
FAU_GEN.1		X						
FAU_STG.1		X						
FCS_CKM.4	X							
FCS_COP.1	X							
FDP_ITC.2				X	X		X	
FDP_RIP.1							X	
FDP_UCT.1					X			
FDP_ACC.1				X				
FDP_ACF.1				X				
FIA_AFL.1				X				
FIA_ATD.1				X				
FIA_SOS.1								X
FIA_UAU.1					X			X
FIA_UID.1					X			X
FMT_MSA.1				X				
FMT_MSA.3				X		X		
FMT_SMF.1		X		X				
FMT_SMR.1				X				
FPT_TST.1				X			X	
FPT_TDC.1					X			
FRU_RSA.1					X			
FTA_SSL.3			X					
FTA.SSL.4							X	

FTA_TSE.1							x	
FTP_TRP.1					x			

Table 12 Correlation of security functional requirement to security objectives

7.3.2. SECURITY FUNCTIONAL DEPENDENCY RATIONALE

SFR Claim	Required dependencies	Dependency met	Rationale
FAU_GEN.1	FPT_STM.1 Reliable time stamps	NO	FPT_STM.1 satisfied by OE.OPERATING_SYSTEM
FAU_STG.1	FAU_GEN.1 Audit data generation	YES	FAU_GEN.1 is satisfied by addressing the corresponding SFR in this Security Target.
FCS_CKM.4	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation]	YES	FDP_ICT.1 is satisfied by addressing the corresponding SFR in this Security Target.
FCS_COP.1	[FDP_ITC.1 Import of user data without security attributes or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction	YES	FDP_ICT.1 and FCS_CKM.4 are satisfied by addressing the corresponding SFR in this Security Target.
FDP_ITC.2	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] [FTP_ITC.1 Inter-TSF trusted channel, or FTP_TRP.1 Trusted path] FPT_TDC.1 Inter-TSF basic TSF data consistency	YES	FDP_ACC.1, FTP_TRP.1 and FPT_TDC.1 are satisfied by addressing the corresponding SFR in this Security Target.
FDP_RIP.1	None	N/A	-
FDP_UCT.1	[FTP_ITC.1 Inter-TSF trusted channel, or FTP_TRP.1 Trusted path] [FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control]	YES	FTP_ITC.2 and FDP_ACC.1 are satisfied by addressing the corresponding SFR in this Security Target.
FDP_ACC.1	FDP_ACF.1 Security attribute-based access control.	YES	FDP_ACF.1 is satisfied by addressing the corresponding SFR in this Security Target.

FDP_ACF.1	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialization	YES	FDP_ACC.1 and FMT_MSA.3 are satisfied by addressing the corresponding SFR in this Security Target.
FIA_AFL.1	FIA_UAU.1 Timing of authentication	YES	FIA_UAU.1 is satisfied by addressing the corresponding SFR in this Security Target.
FIA_ATD.1	None	N/A	-
FIA_SOS.1	None	N/A	-
FIA_UAU.1	FIA_UID.1 Timing of identification	YES	FIA_UID.1 is satisfied by addressing the corresponding SFR in this Security Target.
FIA_UID.1	None	N/A	-
FMT_MSA.1	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions	YES	FDP_ACC.1 ,FMT_SMR.1 and FMT_SMF.1 are satisfied by addressing the corresponding SFR in this Security Target.
FMT_MSA.3	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles	YES	FMT_MSA.1 and FMT_SMR.1 are satisfied by addressing the corresponding SFR in this Security Target.
FMT_SMF.1	None	N/A	-
FMT_SMR.1	FIA_UID.1 Timing of identification	YES	FIA_UID.1 is satisfied by addressing the corresponding SFR in this Security Target.
FPT_TST.1	None	N/A	-
FPT_TDC.1	None	N/A	-
FRU_RSA.1	None	N/A	-
FTA_SSL.3	None	N/A	-
FTA_SSL.4	None	N/A	-
FTA_TSE.1	None	N/A	-
FTP_TRP.1	None	N/A	-

Table 13 SFR Dependencies rationale

7.4. SECURITY ASSURANCE REQUIREMENTS RATIONALE

The ST specifies Evaluation Assurance Level 2. EAL2 was chosen since it is the based upon good commercial development practices. EAL2 provides assurance by means of this full ST and the analysis of SFRs it contains. Due to EAL2 requires testing the TSF and a vulnerability analysis, it gives some security about the resistance of the TOE to penetration attackers with basic attack potential.

The security environment provides physical protection, and the TOE itself offers an extremely limited interface, making it difficult for an attacker to subvert the security policies.

7.4.1.1. SECURITY ASSURANCE REQUIREMENTS EVIDENCE

This section identifies the measures applied to satisfied CC assurance requirements.

Security assurance requirement	Evidence title
ADV_ARC.1 Security architecture description	*GMV GNSS Cryptographic Module – Security Architecture*
ADV_FSP.2 Security-enforcing functional specification	*GMV GNSS Cryptographic Module – Functional Specification*
ADV_TDS.1 Basic Design	*GMV GNSS Cryptographic Module – TOE design*
AGD_OPE.1 Operational user guidance	*GMV GNSS Cryptographic Module – User Guide*
AGD_PRE.1 Preparative procedures	*GMV GNSS Cryptographic Module – User Guide*
ALC_CMC.2 Use of a CM system	*GMV GNSS Cryptographic Module – Configuration Management Processes and Procedures*
ALC_CMS.2 Parts of the TOE CM coverage	*GMV GNSS Cryptographic Module – Configuration Management Processes and Procedures*
ALC_DEL.1 Delivery procedures	*GMV GNSS Cryptographic Module – Secure Delivery Processes and Procedures*
ASE_CCL.1 Conformance claims	*GMV GNSS Cryptographic Module – Security Target*
ASE_ECD.1 Extended components definition	N/A
ASE_INT.1 ST introduction	*GMV GNSS Cryptographic Module – Security Target*
ASE_OBJ.2 Security objectives	*GMV GNSS Cryptographic Module – Security Target*
ASE_REQ.2 Derived security requirements	*GMV GNSS Cryptographic Module – Security Target*
ASE_SPD.1 Security problem definition	*GMV GNSS Cryptographic Module – Security Target*
ASE_TSS.1 TOE summary specification	*GMV GNSS Cryptographic Module – Security Target*
ATE_COV.1 Evidence of coverage	*GMV GNSS Cryptographic Module – Security Target*
ATE_FUN.1 Functional testing	*GMV GNSS Cryptographic Module – Test plan* *GMV GNSS Cryptographic Module – Test Results*
ATE_IND.2 Independent testing - sample	*GMV GNSS Cryptographic Module – Test plan* *GMV GNSS Cryptographic Module – Test Results*

AVA_VAN.2 Vulnerability analysis	*GMV GNSS Cryptographic Module – Test plan* *GMV GNSS Cryptographic Module – Test Results*

Table 14 Security Assurance rationale

8. TOE SUMMARY SPECIFICATION

This section summarizes the security features of the TOE that permit the satisfaction of the security functional requirements describe in section 7.

8.1. TOE SECURITY FUNCTIONS

The security functions performed by the TOE are as follows:

Security Audit

Access Control

Identification and Authentication

Cryptographic key importation

Security Management

Self-tests

8.2. SECURITY AUDIT

The TOE can generate two types of events that report on the status and the status change of the system. The TOE will audit these events and will make them available to the administrator of the TOE who will transfer it to the Crypto Officer.

Internal events: These events are related to a specific state or change of state of the system. They report the start-up and shutdown of the module and all the state changes specified in the Functional Specification document.

Audit Events: These are the events generated by calling audited methods or modifying any audited data object. The Audit events can be separate into **User Events**, that include, for instance, the events related with the user login and logout, and any modification of user attributes, and the **System Events**, which include the result of the requested operations and the internal errors**.** All the auditable events are specified in Table 8.

All the auditable events are stored in a log file into the ***cryptomodule_2*** file. This file its circular, which implies once a certain level of bytes is reached, it will be rewritten from the first line.

The TOE provides the operating system administrator the capability to read all audit dated generated with the TOE via console or text editor. The administrator will have the permission to modify or delete the audit file and will be responsible to transfer it to the corresponding Crypto Officer.

The log traces include the operational system date and time, the type of the event, the event subject identity, the event outcome, and all additional information described in Table 8.

The Security Audit function is designed to satisfy the following functional requirements:

FAU_GEN.1

FAU_STG.1

8.3. ACCESS CONTROL

The TOE employs a role-based access control (RBAC) system to manage its functions. This means that a user's assigned role dictates the functions, information, and modifications they can access. For example, this includes

access to the audit log, user data, and communication path management. If a client with a user role attempts to execute a crypto-officer function, the system will return an error code.

The role-based access control is implemented through two distinct APIs. One API, prefixed with "API," is designed for users with the USER ROLE, and its functions are labeled accordingly (e.g. **API_updatePass**). The other API, prefixed with "API_CO," is specifically for users with the CRYPTO-OFFICER ROLE, and its functions are labeled accordingly (e.g. **API_CO_deleteUser**). This API calls **allow users** to, for example, access to their associated user data in the file system files without direct access to the files.

A comprehensive overview of the various functionalities accessible to different user roles can be found in Table 10.

The Access Control function is designed to satisfy the following functional requirements:

FDP_ACC.1

FMT_MSA.1

FMT_MSA.3

FMT_SMF.1

FMT_SMR.1

8.4. IDENTIFICATION AND AUTHENTICATION

The TOE enforces individual identification and authentication after allowing the performance of any cryptographic or management function. The TOE supports password-based authentication in addition to the role-based access control (RBAC) that allows assign different levels of access to different users based on their associated roles.

The attributes belonging to individual users are the following:

- User Identity (username)
- Password
- Roles

Upon establishing the trusted channel connection (**utilizing TCP sockets encrypted with AES cipher**), the TOE prompts the client for their user identity (username) and corresponding password. Subsequently, the TOE validates the provided credentials by conducting a SHA256 hash function on the input and comparing it against the user data stored in the file system. The file system stores user data, including the user's ID, username, and hashed password. Throughout all cryptographic operations, the user is uniquely **identified by their ID**.

If the provided credentials match the stored ones, the user will be granted access to the system and the appropriate level of access based on its role, as can be seen in Table 10. If the credentials do not match, the user will be denied access.

To enhance the security against brute force attacks, the module incorporates the following measures:

1. **After three failed login attempts**, the system initiates a login timeout. This procedure terminates the connection and remains active until a successful login occurs within the designated timeframe.

Furthermore, the TOE also includes a password policy that will enforce the use of strong and complex passwords. These passwords are the **secrets referenced by the SFRs**, and must adhere to specific criteria, including a **minimum length of eight characters, comprising both uppercase and lowercase letters, at least one digit, and at least one special character, with no spaces permitted**.

The Identification and Authentication function is designed to satisfy the following security functional requirements:

FDP_ACF.1

FIA_AFL.1

FIA_ATD.1

FIA_SOS.1

FIA_UAU.1

FIA_UID.1

FTA_TSE.1

FPT_TDC.1

FTP_TRP.1

8.5. CRYPTOGRAPHIC KEY IMPORTATION

AES KEY IMPORTATION

Prior to the initial configuration, it is essential to import the AES256 key, which is used to establish a secure communication channel. This task is performed by a user possessing Administrator rights. The user is responsible for securely saving the AES key within a file designated as "init_module" located within the file system of the TOE. The access to the “init_module” file is strictly controlled and facilitated exclusively via the Operating System's file management system, therefore only the Administrator of the system may have access to it.

USERS’ KEY IMPORTATION

When it comes to storing associated users' keys - ECDSA Public Key and/or TESLA Root Key -the TOE guarantees secure importation by transitioning to the CSP_STATE. Furthermore, the TOE implements a thorough zeroization process to securely erase any data stored in the module when a CO role user performs the operation or in case of **malicious use** of the module before deletion (e.g. after three failed login attempts when the module is in ***ERROR_STATE***). This zeroization zeroizes **every data in the cryptographic module**, including the ***cryptomodule_2*** and ***cryptomodule_3*** files, but it does not delete the ***cryptomodule_1*** executable. To zeroize the data, first it **is set with all zeros**, and then it is **deleted**. Moreover, the TOE rigorously audits all operations related to the addition, modification, or removal of cryptographic keys via its robust trace system. All the cryptographic operations supported by the TOE can be found in **Table 9 Cryptographic algorithms and operations**

Cryptographic Importation functionality is designed to satisfy the following security functional requirement:

FDP_ITC.2

FCS_COP.1

FCS_CKM.4

8.6. SECURITY MANAGEMENT

Security management encompasses the administration of functionality related to the communication channel, the secure management of cryptographic algorithms, and the handling of secure memory management.

The management functions and its corresponding SFR are noted below:

SFR	TSF Description
FCS_CKM.4	The TSF ensures the appropriate zeroization method is applied when cryptographic keys of the defined algorithms in Table 9 are deleted after requesting it or in case of malicious use of the module (e.g. after three failed login attempts when the module is in ERROR_STATE)
FCS_COP.1	The TSF employs cryptographic algorithms that are officially approved by the European Space Agency (ESA) authority. Each of these algorithms adheres strictly to the specifications outlined in the corresponding standards, as detailed in Table 9 Cryptographic algorithms and operations**.**
FDP_RIP.1	The TSF implements a residual information protection mechanism that zeroizes on demand both memory and file storage.
FDP_UCT.1	The TOE enforces the confidentiality of the transmitted data from/to the client since it uses an AES encryption in CBC mode on all transmitted packets. It also implements a Nonce mechanism to prevent replay attacks.
FRU_RSA.1	The TOE establishes the upper bound of 30 kB for the packet size and 3 Bytes maximum for each packet parameter. Ten minutes are established as maximum time of inactivity. This default values cannot be modified by any user or TOE administrator.
FTA_SSL.3	After exceeding the inactivity period, the TOE closes the current connection and waits for a new connection.
FTA_SSL.4	The TSF provides the capability for user-initiated termination of interactive sessions.
FTA_TSE.1	The TSF rejects any connection if there is already and established connection. It also rejects the connection in case the number of attempts for authentication is exceeded.
FPT_TDC.1	Each user's data within the TOE's file system is uniquely distinguished by associating it with a specific user ID. This data allocation includes the username, the SHA256 hash of the user's password, and records of the most recently updated Root and Public Keys by that user.

Table 15 Functional description of the Security Management SFRs

8.7. SELF-TESTS

After the initial startup, the GMV GNSS Module **checks the integrity** of its executable code and performs a series of **self-tests on its cryptographic functions**. Only upon successful completion of these checks and tests does the module permit the execution of any operational requests.

Additionally, this process of verifying code integrity and performing cryptographic self-tests can be initiated on-demand through the **TSFI self-test function**. This feature is accessible exclusively to users authenticated with the Crypto-Officer role.

Self-Test functionality is designed to satisfy the following security functional requirement:

FPT_TST.1

9. TOE SECURITY FUNCTIONS RATIONALE

SFR	TSF
FAU_GEN.1	Security Audit
FAU_STG.1	Security Audit
FCS_CKM.4	Cryptographic Importation and Security Management
FCS_COP.1	Cryptographic Importation and Security Management
FDP_ITC.2	Cryptographic Importation
FDP_RIP.1	Security Management
FDP_UCT.1	Security Management
FDP_ACC.1	Access Control
FDP_ACF.1	Identification and Authentication
FIA_AFL.1	Identification and Authentication
FIA_ATD.1	Identification and Authentication
FIA_SOS.1	Identification and Authentication
FIA_UAU.1	Identification and Authentication
FIA_UID.1	Identification and Authentication
FMT_MSA.1	Access Control
FMT_MSA.3	Access Control
FMT_SMF.1	Access Control
FMT_SMR.1	Access Control
FPT_TST.1	Self-Tests
FPT_TDC.1	Identification and Authentication and Security Management
FRU_RSA.1	Security Management and Security Management
FTA_SSL.3	Security Management and Security Management

FTA.SSL.4	Security Management and Security Management
FTA_TSE.1	Identification and Authentication and Security Management
FTP_TRP.1	Identification and Authentication

Table 16 TSF Rationale

END OF DOCUMENT