USER GUIDE

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: 28/08/2024
Version: v2.1.7

TABLE OF CONTENTS

LIST OF TABLES AND FIGURES

1. INTRODUCTION

1.1. PURPOSE

The purpose of this document is to ensure that all types of users can operate the GMV GNSS Cryptographic Module in a secure manner according to the certification Common Criteria EAL2 evaluation.

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)
Role	Pre-defined set of rules establishing the allowed interactions between a user and the target of evaluation (TOE)
Target Of Evaluation (TOE)	Set of software, firmware and/or hardware possibly accompanied by guidance, which is the subject of an evaluation.

Table 1 Definitions

1.2.2. ACRONYMS

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

Acronym	Definition
CO	Crypt-Officer Role
UG	User Guide
SFR	Security Functional Requirement
ST	Security Target
TOE	Target of Evaluation

Table 2 Acronyms

2. REFERENCES

2.1. REFERENCE DOCUMENTS

The following documents, although not part of this document, amplify or clarify its contents. Reference documents are those not applicable and referenced within this document. They 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.3 - v2.1.7	Common Criteria for information Technology Security Evaluation. Part 3: Security assurance components	3.1	5	April 2017
RD.4 - v2.1.7	GMV GNSS Cryptographic Module - Security Target	v2.1.7	-	29/01/25
RD.6 - v2.1.7	GMV GNSS Cryptographic Module - Functional Specification	v2.1.7	-	29/01/25
RD.10 -v2.1.7	GMV GNSS Cryptographic Module – Secure Delivery Processes and Procedures	v2.1.7	-	29/01/25

Table 3 Reference Documents

3. USER GUIDE INTRODUCTION

3.1. UG REFERENCE

UG Title	User Guide: GMV GNSS Cryptographic Module
Author	Esther Godoy, Jorge Juan Tejero, Alberto Sendino, Fernando Elena, Teresa Feu Basilio, Enrique Robles Uriel
Date	29/01/25
UG Version	v2.1.7

3.2. TOE REFERENCE

TOE developer	GMV Aerospace and Defence
TOE name	GMV GNSS Cryptographic Module*
Date	29/01/25
TOE version	v2.1.7

**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. OBJECTIVES

The User Guide objectives are to guide every possible user of the GMV GNSS Cryptographic Module, such as end-users, people responsible for maintaining and administering the TOE in a correct manner for maximum security, and by others (e.g., programmers) using the TOE's external interfaces.

The User Guide help ensures that all types of users can operate the TOE in a secure manner, so it is indispensable to follow the guide, as well as to manage warning and errors during operation.

This guide provides the required information to install and deliver securely the cryptomodule. Here it is described how the module will be delivered to the client and how it will be installed.

4. GENERAL DESCRIPTION

The GMV GNSS Cryptographic Module is a comprehensive software suite designed to function as a network device, delivering cryptographic support services to external users connected via socket connections.

In the next sections a comprehensive overview about the different aspects to consider when using the GMV Module will be explained. This will include preparative procedures for installation and initialization, the different security functions for each authorized role, as well as the explanation of the most significant security events and warnings the different users may encounter while using the module.

4.1. IDENTIFICATION AND CHARACTERISATION OF USER ROLES

The module supports two distinct user roles, each encompassing specific security functions and operational procedures when interacting with the module. These roles are assigned to users during their creation within the system, except in the first use of the module since the Default user has crypto-officer role by default, as it requires the privilege to create new users, and set-up the module.

It is important to note that the administrator user type cannot be assigned to new users. This type is exclusively established for the person responsible for installing and initializing the module.

The different authorized role for the module is described below.

- **USER**

The user role "**user**" refers to end-users. Users will be the clients that will use the module regularly. Usually, when we talk about users, we talk about most of the traffic that the module will handle. The end-user is the user with less privileges, they can only perform the cryptographic functions and some own-user management functions, like password changes and manage their own public and root keys. This role is the **default role** set when a new user is created in the module.

- **CRYPTO-OFFICER**

The user role "**crypto officer**" (CO) refers to users with some privileges such as create other users, change passwords, update roles, execute the self-test functions and zeroize the module.

- **ADMINISTRATOR**

The user type "**administrator**" refers to the person that has the responsibility to install and initialize the cryptomodule. The administrator will have to deal with the first interaction with the cryptomodule physically. **This administrator is the crypto officer leader, which means it shall access the same functions than a usual crypto officer. From now on, every crypto officer specification can be also applied to the administrator user type.**

- **MAINTAINER**

The user type "**maintainer**" refers to the person in charge of starting up the module when it goes into ERROR_STATE or crashed and becomes totally unusable. The maintainer will receive the module, prepare it for use and send it back to the customer. Since the maintainer has access to the source code, it could even generate other copies of the files needed for the module to work if the corrupted client files could not be fixed. **The maintainer will have the role of crypto officer on a functional basis, which means it shall access the same functions than a usual crypto officer. From now on, every crypto officer specification can be also applied to the maintainer role.**

The following table summarises the different role privileges.

Role	Create & Manage users	Access to Cryptofunctions
User		✓
Crypto-officer	✓	✓

Table 4 Summary of role privileges

Note: The user types "Administrator" and "Maintainer" have physical access to the TOE for its installation and maintaining operations respectively. Both types have the "Crypto-officer" role.

4.2. TOE MODES OF OPERATION

The modes of operation are related with the **different scenarios** in which a client can perform functions and operations in the cryptographic module. Depending on the mode of operation in which the module is, it will allow the performance of some **specific functions and operations**, as well the access of certain users depending on their user role. The GMV GNSS Cryptographic Module differentiate **three different modes of operation**:

4.2.1. INITIAL CONFIGURATION MODE

The initial configuration mode is the **module factory mode**, without most of the information than the module requires to start a normal functionality (**AES key and IV, access port**). In this mode of operation, only the system administrator shall access using the default user (user: **default** password: **Default123***). The operations allowed in this mode of operation are the **login and the user credential modification**, so the system administrator can include the first secure user, the access port, and the AES keys to cipher the communications between client and module. To see the first initialization process, see **section 3.1** in RD.10 -v2.1.7**.**

After the first initialization, the initial configuration mode **cannot be re-established**, so there is no option to change the cipher keys or the access port. To change these parameters, the cryptographic module shall be **re-manufactured and restored in factory.**

4.2.2. NORMAL MODE

The normal mode is the **most common mode of operation** presented in the module. Is the mode where most of the functions, operations and management can be performed for any user, either user role or crypto officer role. **All** the cryptographic module functions and operations **are allowed with the corresponding role**, and any user can access in this normal mode, so it is important to login them, to manage their actions and restrict the functions that are forbidden for them. To see more about the interaction between client and module in this mode of operation, see **section 3.1** in RD.6 -v2.1.7.

Except for the first initialization, the module will always be in this mode, unless the cryptographic module enters an ***ERROR_STATE*** (where the mode of operation becomes the **error mode** described later). Once the cryptographic module is in this error mode, it will not be able to return to **normal mode** again, so the module **should be taken to factory.**

4.2.3. ERROR MODE

The module enters the error mode of operation when the state is the ***ERROR_STATE***. This state is reached when the module **detects that it is under threat or under attack**, f.e if a malicious user tries to attack the module and fails the authentication a certain number of times, the module consideres this user as an attacker and a danger to the module. This mode of operation exists to **allow a crypto officer user to reset the module and zeroizes** all the information from it, preventing the success of attack.

In this mode of operation, **only** the **crypto officer user role is allowed**, and the user can only log into the module, and execute the zeroization function (**type:2 id:10 of the packet structure**). If an attacker tries to access into the error mode of operation, the module will also record the login attempts,

and as in the previous case, **if the maximum of failed attempts is reached**, the module will be **automatically zeroized**. Once the module is restarted, it becomes unusable, therefore it **shall return to factory.**

4.3. PREPARATIVE PROCEDURES

This section begins by outlining the initial reception of the TOE in accordance with the RD.10 -v2.1.7 document. Subsequently, it delves into the preparation of the operational environment for the TOE. Following that, it provides a detailed description of the necessary installation steps. Lastly, it covers the procedures for starting up the TOE.

4.3.1. RECEPTION OF THE TOE

This guide outlines the secure procedures for receiving the GMV GNSS Cryptographic Module, with the administrator designated as the responsible role for the module's reception. All the specific steps for this process have been elaborated in RD.10 -v2.1.7.

Here are defined the summarized set of **steps** to follow to **receive securely the module**:

1. The client and the distributor shall be agreeing a date for the delivery. The distribution is conducted via **encrypted e-mail**. GMV and the client exchange GPG public keys, and GMV encrypts the email containing the GMV GNSS Cryptographic Module in a zip file.
2. The zip file delivered has the name "**gmv-gnss-cryptographic-module-v2.1.7.zip**". Each release version is represented by a zip file. once the module is imported it shall be on an Ubuntu system. After that, the user can use the command '***unzip gmv-gnss-cryptographic-module-v2.1.7.zip***'. The following files will be seen after unzipping this file:
    - **[Folder] Zip archive labelled "GMV_GNSS_Cryptographic_Module_v2.1.7": This zip contains all the constituent files of the TOE:**
        - **[File] cryptomodule_1.elf:** Executable file specifically designed for operating on the Ubuntu 22.04 platform. It serves as the primary system runner.
        - **[File] cryptomodule_2.bin:** File responsible for maintaining data persistently archived in the system.
        - **[File] cryptomodule_3.bin:** File designated for storing the metadata associated with the general file system module.
        - **[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 Specification.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.
3. The subsequent step involves verifying the module's integrity. To accomplish this, the following command may be utilized for each file: `sha256sum <file_to__verify>`

If the resulting hash from the command matches the hash listed for the file in **hashes_file.txt**, it confirms the integrity of the module. Should there be a discrepancy, indicating a failure in the integrity verification, the administrator is advised to promptly contact GMV for further assistance.

Finally, if the confidentiality and integrity checks have been verified, the module is ready to use.

4.3.2. PREPARATION OF THE OPERATIONAL ENVIRONMENT

This section describes all the steps necessary for the secure preparation of the operational environment. For this preparation the security objectives for the operational environment in should be considered. The following steps shall be considered:

1. Ensure that the applications which use the TOE perform the security checks on the data through the secure channel before transforming it and sending it to the TOE.
2. Verify that the system has at least 2 MB of available storage to accommodate the audit traces.
3. The operating system where the TOE will be allocated, shall be secure and provide reliable timestamps. It is recommended to use Ubuntu 22.04 as the operating system.
4. Ensure the TOE's housing location has implemented proper physical security measures.
5. Finally, the personnel using the TOE shall be trained in its use, to use it in a correct manner.

4.3.3. INSTALLATION OF THE TOE

This guide shows the procedures to install securely the cryptomodule once the module is set in its final location.

The administrator is the role that will install the module physically. Here are defined the **steps** to follow to **install securely the module**:

1. The administrator and the client will agree on a directory to place the GMV GNSS Cryptographic Module files.
2. The administrator will place the GMV GNSS Cryptographic Module files in the folder demanded.
3. The administrator will check if the IP direction the port demanded could be used. By default, the IP is 127.0.0.1 (*localhost*) and the port 8080. If the port couldn't be used, in the setup of the TOE, the port can be changed. See section 4.3.4.
4. The administrator shall import the AES key and AES IV which will be used to encrypt the communication channel with the TOE.

Now the module is ready for the setup.

4.3.4. SETUP OF THE TOE

This guide shows the procedures to initialize securely the GMV GNSS Cryptographic Module once the module has been correctly installed.

1. First, the security measures to meet the security objectives for the operational environment are defined:

The user should implement a method to verify that the data being sent is correctly formatted, for example, an application that checks the proper formatting of the packets before sending them to the module. This application shall be compliance the following security checks:

I. Access control. It may be in a locked room accessible only to the administrator, who is the only one who should have physical access to the TOE.

II. Updates and patches. The application shall be updated to the latest version.

III. Use by trained personnel. Personnel who have read the TOE documentation, understood it, and could additionally receive some training in its use.

The secure channel with the TOE is that communications are encrypted with the AES key stored in the TOE. This key must be imported by the application that will communicate with the TOE. The AES key is on the third line of the data file (cryptomodule_2) and is composed of 32 bytes, while the IV is on the next line and is 16 bytes long. An example is shown in Figure 1, where you can see the key and IV used by the TOE, the first characters are shown, but the others have been covered for security reasons.

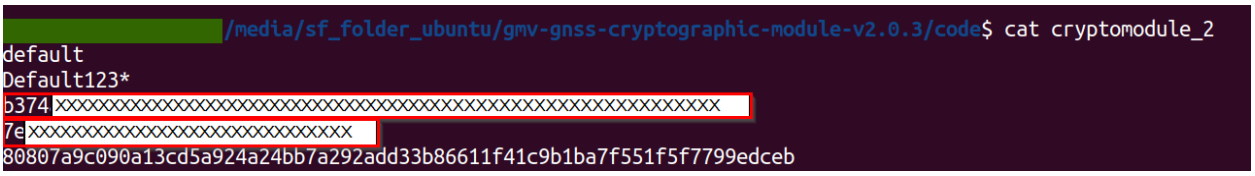

Figure 1 Output data file.

3. To verify the system space the command "*df -h /*" can be used, and the free space will be given by the column "*Avail*", as marked in red in Figure 2. It must be verified that the value obtained is greater than 2M apart from the module size.

Figure 2 Command to check OS available space.

4. To verify if the operating system is secure and provide reliable timestamps, the specific version of Ubuntu will be checked using, for example, the command "*lsb_release -a*" as shown in Figure 3. In this figure, the value marked in red is the version value. Once obtained, it should be verified that it still has support, and repositories and packages should be updated using the command "*sudo apt update && sudo apt upgrade*". After doing this, the "*date*" command will be used to verify that the clock displays the current date and time.

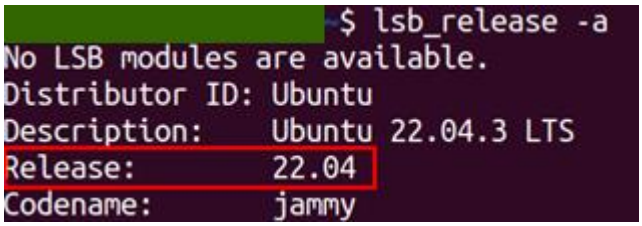

Figure 3 Command to view the OS version.

5. To ensure physical security, the device should be housed in an area where there are security personnel controlling access. Additionally, it could be in a locked room to prevent unauthorized personnel from tampering with it. Also, access control policies could be defined. In these policies, the administrator shall be the only one with physical access to the TOE. Finally, the TOE should be part of the logical measures, if they are already in place, such as an IDS or IPS as a minimum, although a firewall could be added in addition. If they are not implemented, the ideal would be to add an IDS to detect and an IPS to respond to intrusions.
6. To ensure that personnel are trained in its use, they should read all documentation, understand it and, if necessary, undergo training.

The administrator is the role that will initialize the module. Here are defined the **steps** to follow to **initialize securely the module**:

1. The administrator will run the binary (./*cryptomodule_1*) initializing the module.
2. After the self-tests passed, the administrator will login the module with the default credentials sending the corresponding login packet (See section 5.1). This default user has the Crypto-Officer role by default.
    - Username: *default*
    - Password: *Default123**
2. The administrator shall change the default username (*default*) and password (*Default123**) to another one with the corresponding *CO update username* and *CO update password* packets (See section 6.1)

3. After the default username and password is changed, the client shall verify the TOE version of the module. To perform it, first send the **login packet** and then the **show version packet**, as described in RD.6 - v2.1.7.

Once this process is done, the GMV GNSS Cryptographic Module is ready to use with the new credentials associated to the CO role.

5. USER ROLE SPECIFIC DESCRIPTION

5.1. USER ROLE - SECURITY FUNCTIONS

Users will access the module through the socket via data packets. For more information about data packets and how to invoke them read RD.6 - v2.1.7. Every packet gives access to security functionalities. Each user will have access to several security functionalities depending on its role.

Note that every communication between the module and the users will be ciphered with the AES algorithm.

The following table describes all the security interfaces associated to the corresponding packet type and ID of the user (end-user).

TSF	Type	ID	TSFI	Parameters
Cryptographic Operations	01	01	TSFI HMAC verification	Key, message, sign
		02	TSFI CMAC verification	Key, message, sign
		03	TSFI ECDSA verification	Message, sign, curve
		04	TSFI SHA hash	Message, mode
		05	TSFI Merkle tree verification	Intermediate nodes, lead node, root node, node ID
		06	TSFI TESLA function	Key, GST, alpha
Key management	02	01	TSFI users update own RK	Root key
		02	TSFI users update own PK	Public key
		03	TSFI users delete own RK	None
		04	TSFI users delete own PK	None
Users management	03	01	TSFI users change own password	Old password, new password
Identification and Authentication	04	01	TSFI login	Username, password
		02	TSFI logout	None
Module state and information	05	01	TSFI module state information	None
		02	TSFI module general information	None

Table 5 User General Security Interfaces

For more information about the TSFIs, see RD.6 - v2.1.7.

6. CRYPTO-OFFICER ROLE SPECIFIC DESCRIPTION

6.1. CRYPTO-OFFICER ROLE - SECURITY FUNCTIONS

Crypto officers will access the module through the socket via data packets. For more information about data packets and how to invoke them read RD.6 - v2.1.7.

Note that every communication between the module and the crypto officers will be ciphered with the AES algorithm.

The following table describes all the security interfaces associated to the corresponding packet type and ID of the crypto officer.

TSF	Type	ID	TSFI	Parameters
Cryptographic Operations	01	01	TSFI HMAC verification	Key, message, sign
		02	TSFI CMAC verification	Key, message, sign
		03	TSFI ECDSA verification	Message, sign, curve
		04	TSFI SHA hash	Message, mode
		05	TSFI Merkle tree verification	Intermediate nodes, lead node, root node, node ID
		06	TSFI TESLA function	Key, GST, alpha
Key management	02	05	TSFI CO update RK	Username, root key
		06	TSFI CO update PK	Username, public key
		07	TSFI CO delete RK	Username
		08	TSFI CO delete PK	Username
Users management	03	02	TSFI CO create user	Username, password, role
		03	TSFI CO delete user	Username
		04	TSFI CO update username	Old username, new username
		05	TSFI CO update password	Username, new password
		06	TSFI CO update role	Username, new role
Identification and Authentication	04	01	TSFI login	Username, password
		02	TSFI logout	None
Module information	05	01	TSFI module state information	None
		02	TSFI module general information	None
Modules Security Management	02	09	TSFI self-test function	None
		10	TSFI module zeroization	None

Table 6 Crypto-Officer General Security Interfaces

For more information about the TSFIs, see RD.6 - v2.1.7**.**

7. PROCEDURES IN RESPONSE OF SECURITY EVENT

In case some of the next security events have been detected, the system will leave a trace in the log file, and the user will have to respond consequently to the description. For more information of the response codes read RD.6 - v2.1.7.

7.1.1. ERROR STATE TRACES

In case the TOE enters in error state, it will leave a **"208 ERROR"** trace, and the TOE will restrict the use of the module to only crypto-officers. In this case, the TOE should be zeroized by a crypto-officer, and send to a maintainer, so the TOE can be re-initialized in a secure state. In case the module is in error state, and a maintainer restarts the TOE without re-initializing it, the module will generate a **"103 PREVIOUS_ERROR_STATE"** trace and the TOE will keep being in error state.

7.1.2. SELFT-TEST FAILED TRACES

If the self-tests fails, the TOE will generate a "**330 SELF_TEST_NOT_PASSED"**, as consequence the TOE will interpret it as his integrity have been compromised, and enter in error state, the response to this event is similar to **9.2.1 response**.

7.1.3. NOT SUFFICIENT PERMISSION TRACE

In case a persistence function is called without the sufficient permission, it means that the first layer of security has been bypassed, this applies to crypto function and persistence functions, in this case depending on the gravity, the module will return an error code, or enter error state in which the response is similar to **9.2.1 response**. (traces like **480**, or **692**)

7.1.4. MODULE ZEROIZATION TRACE

If the module is zeroized, it will leave a **"951"** trace, in which case, the module needs to be re-initialized again by a maintainer to be useful.

7.1.5. MODULE CRASH

In case the module crash, a maintainer will have to restart the module, in case after the restart, the module is not working correctly, or is in error state, the module will have to be re-initialized.

END OF DOCUMENT