THALES ALENIA SPACE

ESPRIT/COMHUB FPGA

FIPS 140-3 Non-Proprietary Security Policy

Written by	***Responsibility*** + handwritten signature if no electronic workflow tool
G.BRICAS	FPGA designer
Verified by	
F.MOUNIER	FPGA Designer
G.MARCH	FPGA requirements specification writer
F.SEISDEDOS	Hardware Architect
C.BOUE	COMHUB Technical responsible
Approved by	
Quality Team	Quality assurance responsible
M-H.SILLY	Equipment project manager

OPEN

Table of Contents

List of Tables

List of Figures

1 General

1.1 Overview

This document defines the Security Policy for the COMHUB, hereafter denoted the equipment. The equipment COMHUB is responsible of the communication between the Lunar Gateway and equipment on the lunar ground. It integrates an FPGA which contains the cryptographic functions.

1.2 Security Levels

Section	Title	Security Level
1	General	1
2	Cryptographic module specification	1
3	Cryptographic module interfaces	1
4	Roles, services, and authentication	1
5	Software/Firmware security	1
6	Operational environment	1
7	Physical security	1
8	Non-invasive security	N/A
9	Sensitive security parameter management	1
10	Self-tests	1
11	Life-cycle assurance	1
12	Mitigation of other attacks	N/A
	Overall Level	1

Table 1: Security Levels

2 Cryptographic Module Specification

2.1 Description

Purpose and Use:

The aim of the Cryptographic Module, is to accompany the processor to be used as a communication hub with the distant equipment. The Cryptographic Module provides ciphering, deciphering and authentication services for message transmission. The FPGA is interfaced to the processor through a local bus, to a Non

Volatile Memory (NVM) containing boot code and Application Software (ASW), a storage Safe Guard Memory (SGM) for the storage of ASW context information, a Platform (PF) Bus, and internal registers. It is also responsible of communication via external flow channels and the microprocessor.

A specific block in the FPGA is dedicated to the cryptographic functions. It manages the secret elements storage in an external MRAM, and the cryptographic processing on frames. Two types of data can be transmitted through the FPGA. Space module internal flow (not encrypted) and space module external flow (may be encrypted or authenticated). It is able to encrypt/decrypt messages from 128 bytes to 2048 bytes.

Module Type: Hardware

Module Embodiment: Multi-Chip Embedded

Module Characteristics:

Cryptographic Boundary:

The COMHUB is identified as physical perimeter and the FPGA with Secret Element MRAM memory as the logical perimeter.

Tested Operational Environment's Physical Perimeter (TOEPP):

The cryptographic module (FPGA, MRAM) is integrated on COMHUB processing board allowing communication with other components of the space module as depicted on the following pictures:

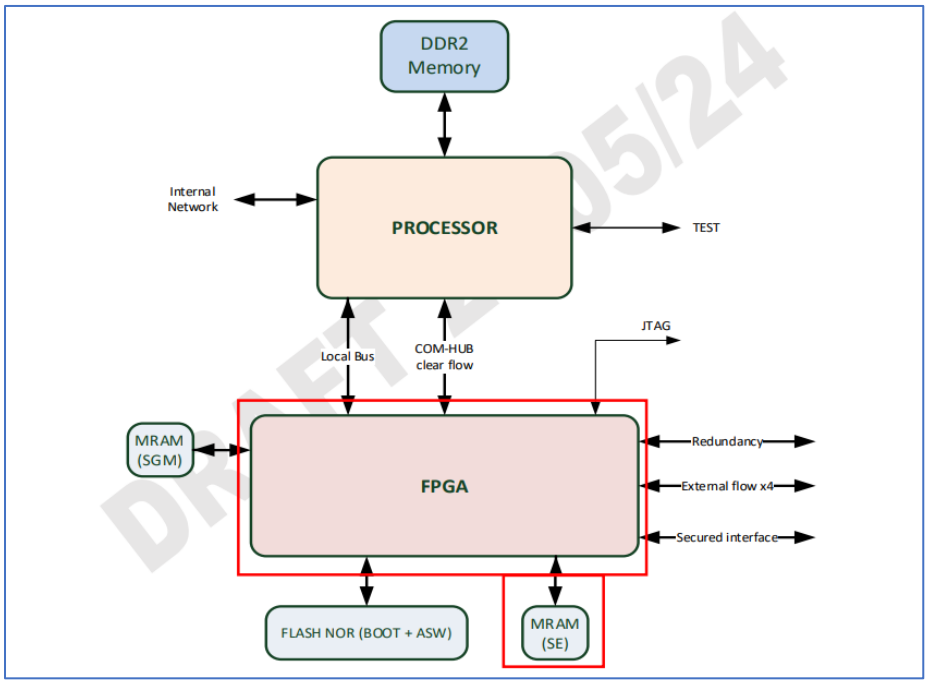


Figure 1 : Comhub Synoptic

Cryptographic boundary

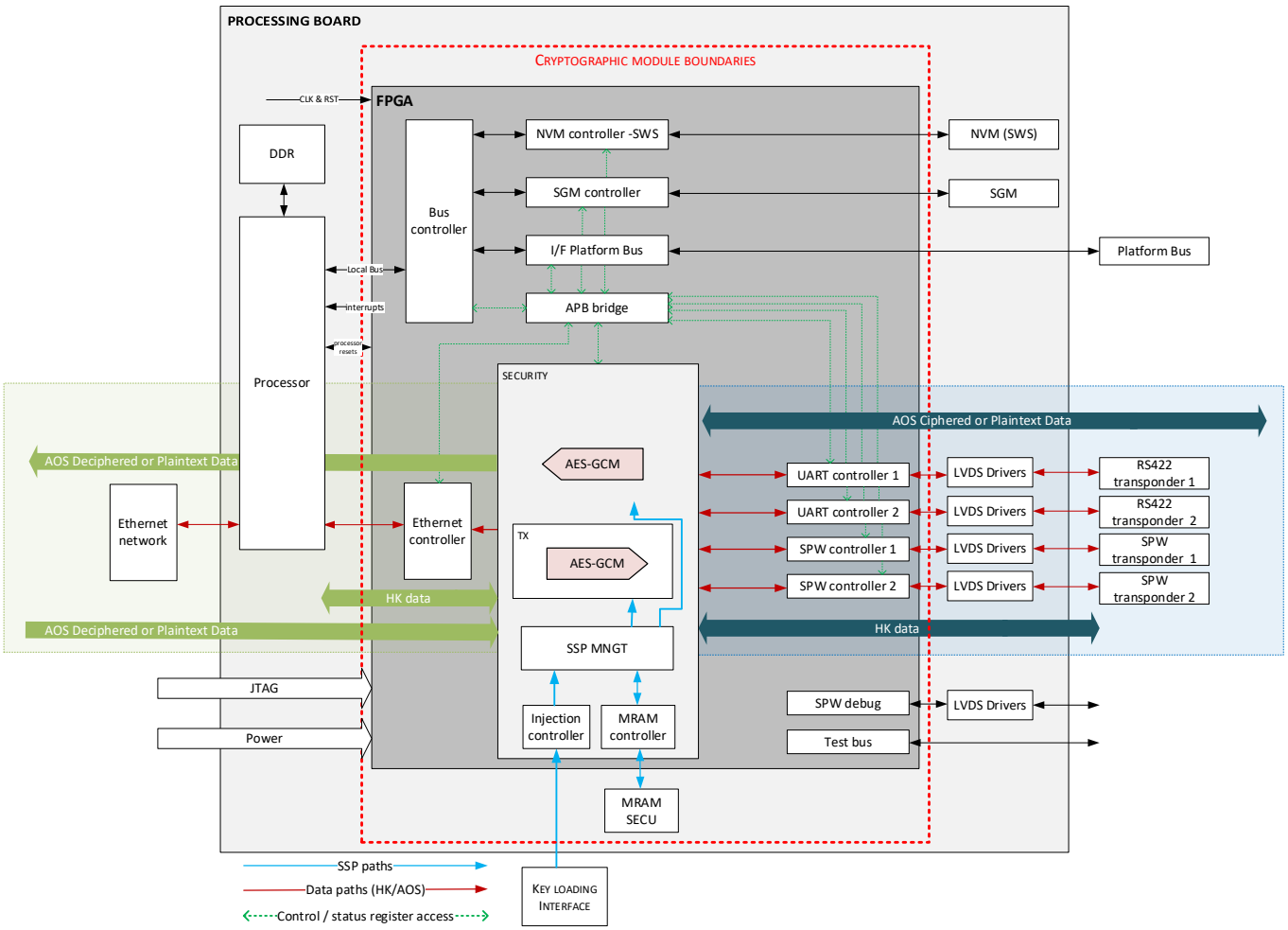


Figure 2: Block Diagram

Only the FPGA and MRAM implements FIPS-approved security functions. The other COMHUB's board components, such as the microprocessor software, are not included in the cryptographic module boundary. The following photography present the MRAM and the FPGA that constitutes the cryptographic module:

2.2 Tested and Vendor Affirmed Module Version and Identification

Tested Module Identification – Hardware:

Model and/or Part Number	Hardware Version	Firmware Version	Processors	Features
FPGA Microsemi RT4G150-x CG1657y with MRAM 3Dplus 3DMR8M08VS8666	N/A	5.10	N/A	x = Blank (Standard speed) or 1 (15 % faster than the standard speed), y = E (Advanced Traditional Space) or B (Entry Level Traditional Space) or PROTO (Prototyping)

Table 2: Tested Module Identification – Hardware

Tested Module Identification – Software, Firmware, Hybrid (Executable Code Sets):

N/A for this module.

Tested Module Identification – Hybrid Disjoint Hardware:

N/A for this module.

Tested Operational Environments - Software, Firmware, Hybrid:

OPEN

N/A for this module.

Vendor-Affirmed Operational Environments - Software, Firmware, Hybrid:

N/A for this module.

2.3 Excluded Components

No components are excluded.

2.4 Modes of Operation

Modes List and Description:

Mode Name	Description	Type	Status Indicator
Approved Mode	The cryptographic module is operating as a compliant FIPS 140-3 module	Approved	The module always operates in the approved mode

Table 3: Modes List and Description

The FPGA implements two algorithms for operational data emission and reception. A FIPSapproved determinist IV generation mechanism is used to ensure the key/IV uniqueness for frame ciphering/deciphering. The dynamic part of this IV generator is also used as an anti-replay check mechanism.

Degraded Mode Description:

After self-test operations, if the result is not OK, the FSM goes in error state to set the status of the failure of the self-test. Then it entries in degraded state.

To exit this state, the equipment shall be powered-off and powered-on.

In this state, the different algorithms and security functions are disable and all data entering the module to be encrypted or decrypted are discard.

It only maintains the internal flow active. The data of this flow cannot get out of the COMHUB board.

The following services are still available:

- Show status.
- Version.

The following services are not available:

- Encryption.
- Decryption.

- Authentication.
- Bypass mode.
- Self-tests.
- Configuration.

When one of these services is requested when an external frame is received, the frame is automatically discarded. No access to the MRAM are authorized.

2.5 Algorithms

Approved Algorithms:

Algorithm	CAVP Cert	Properties	Reference
AES-GCM	A7175	Direction - Decrypt, Encrypt IV Generation - External Key Length - 256	SP 800-38D
AES-GMAC	A7175	Direction - Decrypt, Encrypt IV Generation - External Key Length - 256	SP 800-38D

Table 4: Approved Algorithms

256 keys are available in MRAM for the cryptographic functions. The length of the plaintext data to be ciphered/ cipher text to be ciphered is contained between 98 bytes and 2018 bytes, for a total frame length (header + data + trailer + optional FEC) comprised between 128bytes and 2048 bytes The IV generation uses a deterministic construction.

Vendor-Affirmed Algorithms:

N/A for this module.

Non-Approved, Allowed Algorithms:

N/A for this module.

Non-Approved, Allowed Algorithms with No Security Claimed:

N/A for this module.

Non-Approved, Not Allowed Algorithms:

N/A for this module.

2.6 Security Function Implementations

Name	Type	Description	Properties	Algorithms
Authenticated encryption	BC-AuthEncrypt	encryption of an entry plaintext frame	Publications:[IG C.H]	AES-GCM: (A7175)
Authenticated decryption	BC-AuthDecrypt	decryption of an entry ciphertext frame	Publications:[IG C.H]	AES-GCM: (A7175)
Rx Authentication	MAC	MAC Verification	Publications:[IG C.H]	AES-GMAC: (A7175)
Tx Authentication	MAC	MAC generation	Publications:[IG C.H]	AES-GMAC: (A7175)

Table 5: Security Function Implementations

2.7 Algorithm Specific Information

N/A

2.8 RBG and Entropy

N/A for this module.

N/A for this module.

2.9 Key Generation

N/A

2.10 Key Establishment

N/A

2.11 Industry Protocols

N/A

3 Cryptographic Module Interfaces

3.1 Ports and Interfaces

Physical Port	Logical Interface(s)	Data That Passes
Ethernet interface	Data Input Data Output Control Input Control Output	AOS frames, HK frames.
Local bus	Data Input Data Output Control Input Control Output Status Output	Platform bus, Processor context data, Processor boot & flight software, Status register, Control registers, End of access signal, enable next accesses.
External flow bus (UART) Interface	Data Input Data Output Control Input Control Output	AOS frames, HK frames, Enable signal for LVDS drivers.
External flow bus (SpW) Interface	Data Input Data Output Control Input Control Output	AOS frames, HK frames, Enable signal for LVDS drivers.
Key loading interface	Data Input Control Input Status Output	key injection (DS-102 protocol), Key loader presence and DS102 interface clocking, SSP injection success/failure status.

Physical Port	Logical Interface(s)	Data That Passes
Platform bus	Data Input Data Output Control Input	Platform TMTC, Bus address and address parity.
NVM (SWS) interface	Data Input Data Output Control Input	Processor boot and flight software, Ready/busy signals.
SGM interface	Data Input Data Output	Processor context data.
JTAG	Data Input	FPGA configuration (in factory mode only, deactivated before entering in production).
Global clocks & resets	Control Input	Clocking and reset signals.
Processor reset	Control Input Control Output	Reset requests from/to external processor.
1V2 interface	Power	N/A
2V5	Power	N/A
3V3	Power	N/A

Table 6: Ports and Interfaces

To be noted:

- In key loading mode the Operational Links (IN/OUT) are locked.
- During Self-tests, the operational links (IN/OUT) and the key loading link are locked.
- In error mode (degraded mode), the RX and TX cryptographic function cannot be used.
- In error mode (degraded mode), the MRAM bus is also locked.

4 Roles, Services, and Authentication

4.1 Authentication Methods

N/A for this module.

The module does not support any authentication method.

4.2 Roles

Name	Type	Operator Type	Authentication Methods
CO	Role	CO	None
User	Role	User	None

Table 7: Roles

The Cryptographic Officer role loads or erase SSPs in COMHUB module (in MRAM) while the User role can manage the FPGA configuration, monitor the status and access to cryptographic services.

Cryptographic Officer and User cannot be concurrent operators. The role attribution is defined at COMHUB startup. If the key loader is connected at power on, the Cryptographic Officer role is attributed, whose services are accessible through the key loading interface. If the key loader is disconnected at power on, the User role is attributed. The COMHUB has to be shut-down and powered on to swap from one role to another.
No further authentication mechanism are required to gain access to these roles.

The module does not support concurrent operators: in key loading mode, there is only one key loading interface supporting only one operator at a time while in nominal mode, the microprocessor is the only device able to take the user role.

4.3 Approved Services

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Key loading	Used to upload the Keys and Sercret Elements in MRAM. To upload the secret elements, the CO has to follow these steps: - 1: Connect the	CTS(Clear To Send) signal	Traffic keys	N/A	None	CO - Traffic key: W

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
	Key loader on DS-102 interface and turn on the board. - 2: The CO waits the CTS (Clear To Send) toggling. - 3: The CO injects one secret element. Loop to step2 until all secret elements have been successfully uploaded.					
Zeroization	Used to zeroize the Keys in MRAM. To zeroize the keys, the CO has to follow these steps: - 1: Connect the Key loader on DS-102 interface and turn on the board. - 2: The CO waits the CTS (Clear To Send) toggling. - 3: The CO injects one secret	CTS(Clear To Send) signal	data field = 0	N/A	None	CO - Traffic key: Z

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
	element(data field = 0). Loop to step2 until all secret elements have been successfully uploaded.					
Authenticated_encryption	Used to perform the authentication (MAC generation) and encryption of an entry plaintext frame and AAD using AES-GCM with the desired 256-bit key.	Internal registers status and ciphertext frame	plaintext frame	encrypted and authenticated frame	Authenticated encryption	User - Traffic key: E
Authenticated decryption	Used to perform the authentication (MAC verification) and the decryption of an entry ciphertext frame and AAD using AES-GCM with the desired 256-bit key.	Internal registers status and plaintext frame	ciphertext frame			
Authenticated_decryption	Used to perform the authenticatio	Internal registers status	authenticated, and	plaintext frame	Authenticated decryption	User -

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
	n (MAC verification) and the decryption of an entry ciphertext frame and AAD using AES-GCM with the desired 256-bit key.	and plaintext frame	ciphered frame			Traffic key: E
Rx Authentication (MAC Verification)	Used to verify the authentication tag of an entry frame, using AES-GMAC with the desired 256-bit key and AAD	Internal registers status and tag output in frame	authenticated frame	plaintext frame	Rx Authentication	User - Traffic key: E
Tx Authentication (MAC generation)	Used to generate the authentication tag for an output frame, using AES-GMAC with the desired 256-bit key and AAD	Internal registers status and tag output in frame	plaintext frame	authenticated frame	Tx Authentication	User - Traffic key: E
Show Status	Used to retrieve general and current information of the cryptographic module. The user can read	Bus local reading registers operation	register address	register value	None	User

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
	any register from the local bus interface.					
Show Version	Used to retrieve the version of the FPGA firmware and the cryptographic module ID (module ID and Vendor ID). These are accessible through read access on the Local bus.	Bus local reading registers operation	version register	version	None	User
Key loading pre-operational self-test	Initiate pre-operational self-test by reset when key loader is connected	CTS(Clear To Send) signal	N/A	N/A	Authenticated encryption Authenticated decryption	CO
Pre-operational self-test	Initiate pre-operational self-test by reset when key loader is disconnected.	Internal registers status	N/A	N/A	Authenticated encryption Authenticated decryption	User
ByPass mode	Used to transfer unsecured frames through the cryptographic module.	Internal registers status and Clear frame	unsecured frames	unsecured frames	None	User

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Configuration	Used to configure cryptographic sessions and choose the cryptographic mode.	Bus local reading registers operation	session parameters	N/A	None	User
HK transmission	Transmission of HK frames from/to Ethernet interface to/from transponders. HK frames are transmitted unconditionally and without modification.	Internal registers status and Clear frame	HK frames	HK frames	None	User

Table 8: Approved Services

'Show version' service provides to the user the mean to identify the version of the identifier of the module. This is done by reading the following registers through the local bus:

- Version register (address: 0x---00010). The value of the register corresponds to the FPGA bitstream version. Expected value: 0x0005000A
- TME36 (0x---0D18C): This corresponds to the vendor ID and module ID. Expected values: 0x10400000 (for TAS/COMHUB 1) and 0x10800000 (for TAS/COMHUB 2).

4.4 Non-Approved Services

N/A for this module.

<Text>

4.5 External Software/Firmware Loaded

N/A

4.6 Bypass Actions and Status

The bypass capability of the cryptographic module is enabled on a frame-by-frame basis, depending on the cryptographic modes of the session open on the considered transponder.

- **No security**: full bypass.
- **Authentication only**: data used for authentication, but transmitted frame bypasses the ciphering/deciphering module.
- **Authentication & encryption**: data used for authentication and transmit the ciphered/deciphered data outputted by the AES-GCM algorithm.

The bypass mode service is authorized to user role only. The bypass capability is used to transmit data from the data input interface toward the data output interface bypassing the authentication and ciphering processing provided by AES-GCM/GMAC algorithm. This bypass capability is enabled for a specific communication channel when a 'no security' session is open. In degraded mode, the cryptographic and bypass channels are locked.

A frame checker is implemented at the output of the Rx and Tx frame processing channels. This checker provides a double check to avoid the transmission of frames routed through the wrong link (crypto, bypass, internal flow). This Checker verifies the cryptographic mode of the session corresponding to the interface on which the frame is received/transmitted and ensures that the frame is routed through the right internal link (crypto, bypass, internal flow). If this cryptographic mode and the link do not match, the frame is discarded.

5 Software/Firmware Security

5.1 Integrity Techniques

N/A

5.2 Initiate on Demand

N/A

5.3 Additional Information

The bitstream is loaded in the FPGA during the ground phase (pre-flight factory mode), using the JTAG connector. The JTAG programming capability is definitively disabled before delivery by activating the One-Time-Programmable (OTP) feature of the FPGA RTG4.

6 Operational Environment

6.1 Operational Environment Type and Requirements

Type of Operational Environment: Non-Modifiable

How Requirements are Satisfied:

Once the bitstream is written into the OTP memory, it becomes the permanent configuration for the FPGA. The design is fixed for the lifetime of the FPGA. Therefore, there is no reloading of the bitstream from an external source (like an external Flash memory) or from an internal reprogrammable memory.

7 Physical Security

7.1 Mechanisms and Actions Required

Mechanism	Inspection Frequency	Inspection Guidance
The COMHUB module is composed of production grade materials that include standard passivation techniques.	N/A	N/A

Table 9: Mechanisms and Actions Required

The COMHUB module is composed of production grade materials that include standard passivation techniques.

7.7 Additional Information

The processing board (PCB embedding the cryptographic module) dimensions are the followings: L=32cm / l=26.5cm / h=3cm.

8 Non-Invasive Security

N/A

9 Sensitive Security Parameters Management

9.1 Storage Areas

Storage Area Name	Description	Persistence Type
MRAM	3Dplus 3DMR8M08VS8666	Static

OPEN

Table 10: Storage Areas

No roles can verify or read the content of the storage MRAM in any way.

9.2 SSP Input-Output Methods

Name	From	To	Format Type	Distribution Type	Entry Type	SFI or Algorithm
Key Injection	CO	CM	Plaintext	Manual	Electronic	

Table 11: SSP Input-Output Methods

To import SSP, the CO has to follow these steps:

1. Connect the Key loader on DS-102 interface and turn on the board.
2. Wait the CTS (Clear To Send) toggling.
3. Inject one SSP.
4. Loop to step2 until all SSP have been successfully uploaded.

During SSP injection, all keys are associated to only one entity, so can be only used in AES-GCM and AES-GMAC cryptographic functions.

The secret elements cannot be read or extracted from the physical perimeter. No roles can verify or read the content of the storage MRAM in any way.

9.3 SSP Zeroization Methods

Zeroization Method	Description	Rationale	Operator Initiation
Zeroization	Replace keys by null value using key loader.	The secret elements can't be read or extracted from the physical perimeter. No roles can verify or read the content of the storage MRAM in any way.	To zeroize the keys stored in MRAM, a cryptographic officer turn-off the equipment, connect a key loader, turn-on the equipment and load keys with value=0.

Table 12: SSP Zeroization Methods

The 256 keys stored in MRAM can be zeroized.
To zeroize SSPs stored in MRAM, a cryptographic officer turns off the equipment, connects a key loader, turns on the equipment and loads keys with value=0.
More precisely, the CO has to follow the following procedure:

1. Connect the Key loader on DS-102 interface and turn on the board.
2. Wait the CTS (Clear To Send) toggling.
3. Inject one secret element (data field = 0).
4. Loop to step2 until all SSPs have been successfully zeroized.

The equipment can be power-off to warranty the SSP compromise.

9.4 SSPs

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
Traffic key	AES key to encrypt/decrypt and/or authenticate transmission/reception flows. There are 256 instances of the key, depending on the module you are communicating with.	256 - 256	AES - CSP	Externally generated		Authenticated encryption Authenticated decryption Rx Authentication

Table 13: SSP Table 1

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
Traffic key	Key Injection	MRAM:Plaintext	No	Zeroization	

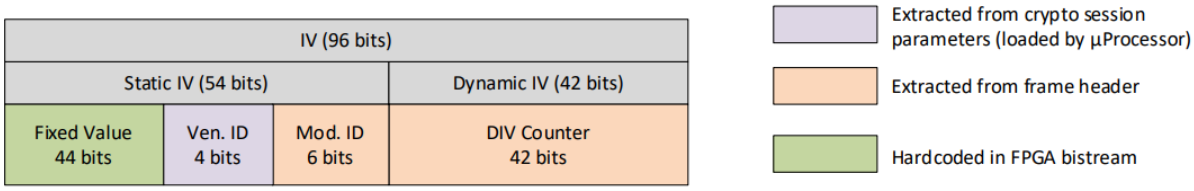
Table 14: SSP Table 2

Additional Information

AES-GCM IVs are a concatenation of 4 fields: a fixed IV, a vendor ID, a module identifier and a dynamic IV counter managed in the security module. No RBG is used. For both RX and TX IV construction, the fixed IV uses a value hardcoded in the FPGA bitstream.

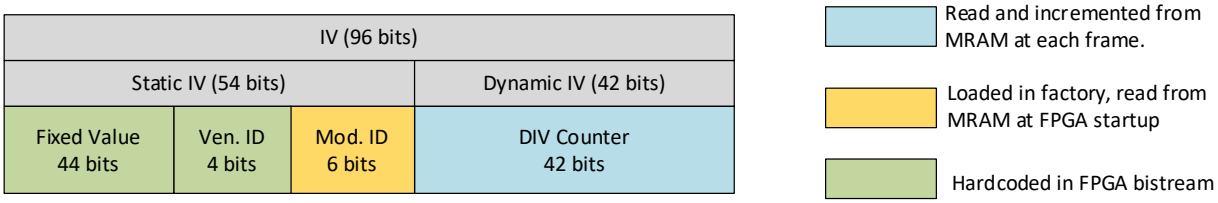
RX IV construction:

- The **vendor ID** is extracted from the cryptographic session parameters (configured from the microprocessor);
- The **module ID** and **dynamic IV counter** are extracted from the frame header.



TX IV construction:

- The **vendor ID** uses a value hardcoded in the FPGA bitstream;
- The **module ID** is read from MRAM at FPGA startup. The module ID can only be loaded in MRAM in factory mode. It is loaded with a different value for each module instance;
- The **dynamic IV counter** is retrieved and incremented from the MRAM matching the key on use.



10 Self-Tests

10.1 Pre-Operational Self-Tests

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details
Bypass clear configuration	N/A	KAT (output in clear)	Bypass	Key loading pre-operational Self-Test: CTS signal (stay high in case of failure), pre-operational normal self-test: RX/TX crypto autotest registers	Configure the mode with "no security", check that data are output in clear
Bypass Authentication only	Key length=256b	KAT (MAC only)	Bypass	Key loading pre-operational Self-Test: CTS signal (stay high in case of failure), pre-operational normal self-test: RX/TX crypto autotest registers	Configure the mode with "Authentication only" and hardcoded parameters (Key, IV, ADD), check that output data match the reference data

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details
Bypass Authenticated encryption	Key length=256b	KAT (MAC + data)	Bypass	Key loading pre-operational Self-Test: CTS signal (stay high in case of failure), pre-operational normal self-test: RX/TX crypto autotest registers	Configure the mode with "Authenticated encryption" and hardcoded parameters (Key, IV, ADD), check that output data match the reference data

Table 15: Pre-Operational Self-Tests

When the COMHUB module is powered-on in key loading mode, the FPGA runs the “key loading pre-operational self-test” made of the following steps:

- RX and TX pre-operational CAST on AES-GCM algorithm.
- RX and TX pre-operational bypass test.

If all the tests are successful, the access to key loading and zeroization services is granted, else, the FPGA transit to the INJ SELF-TEST ERROR mode disabling all services (CTS signal remains high).
As the Local Bus interface is disabled in key loading mode, the only self-test success/failure status is the CTS signal (stays high in case of failure).

When the COMHUB module is powered-on in nominal mode (key loader unplugged), the FPGA runs a pre-operational self-test made of the following steps:

- Secret element self-test: verify the integrity of the 256 traffic keys stored in MRAM through CRC computation.
- RX and TX pre-operational CAST on AES-GCM algorithm.
- RX and TX pre-operational bypass test.

The RX and TX pre-operational tests are run in parallel.
If all RX and TX self-tests are successful, the access to user services is granted, else, the FPGA transit to SELF-TEST ERROR state where self-test errors are reported before before reaching the degraded mode of operation where all RX and TX ciphering, authentication and bypass capabilities are disabled.
The output status of this self-test is reported in the ‘RX Crypto Autotest’ and ‘TX Crypto Autotest’ registers. These registers can be read by the processor through the local bus (show status service).

Pre-operational self-tests are composed of:

1. **Secret element self-test**
   - Reads the 256 cryptographic keys stored in MRAM.
   - The integrity of each key is checked through CRC.
   - If all individual CRC checks are ok, a global CRC is computed on the 256 keys and stored in a dedicated register accessible through the read status service (TME2).
1. **RX/TX CAST**
   - Configure the AES-GCM algorithm with hardcoded parameters: Key, IV, AAD.
   - Send data to the crypto brick (ciphertext for RX, plaintext for TX) and MAC.
   - Wait the end of the encryption and authentication process.

- Verify that the verify MAC operation (check or generate) ended successfully.
- Compare (bit to bit comparison) the output data (plaintext for RX, ciphertext for TX) with expected golden value (hardcoded).

2. RX/TX bypass self-test

a) Bypass clear configuration – Bypass mode:
- Activate ByPass Mode.
- Send data to AES-GCM/GMAC function.
- Check output data match the reference data (output data equal to input data, bit to bit comparison).

b) Bypass Authenticated encryption – Switch to Authentication only mode:
- Deactivate ByPass mode.
- Configure AES-GMAC algorithm with hardcoded parameters: Key, IV, AAD.
- Send data (ciphertext for RX, plaintext for TX) and MAC.
- Check output data match the reference data (ciphertext for RX, plaintext for TX, bit to bit comparison).
- Check the generated MAC match the expected MAC (bit to bit comparison).

a) Bypass Authentication only – Switch to Authenticated encryption mode:
- Configure AES-GCM algorithm with hardcoded parameters: Key, IV, AAD.
- Send data (ciphertext for RX, plaintext for TX) and MAC.
- Check output data match the reference data (plaintext for RX, ciphertext for TX, bit to bit comparison).
- Check the generated MAC match the expected MAC (bit to bit comparison).

c) Bypass clear configuration – Switch back to bypass mode:
- Activate ByPass Mode.
- Send data to AES-GCM/GMAC function.
- Check output data match the reference data (output data equal to input data).

10.2 Conditional Self-Tests

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
AES-GCM encryption	Key length=256b	KAT	CAST	Key loading pre-operational Self-Test: CTS signal (stay high in case of failure), pre-operational normal self-test: TX crypto autotest registers	Encrypt and tag computation	Automatically done at startup

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
AES-GCM decryption	Key length=256b	KAT	CAST	Key loading pre-operational Self-Test: CTS signal (stay high in case of failure), pre-operational normal self-test: RX crypto autotest registers	Decrypt and tag verification	Automatically done at startup
Conditional bypass test	Key length=256b	KAT	Bypass	'Anomaly' registrer through 'Show Status service""	From exclusive bypass to exclusive crypto service, check the cryptographic operation work	When opening a new session
Bypass integrity test	Key length=256b	MAC verification & generation	Bypass	'Anomaly' registrer through 'Show Status service""	Verify the MAC on the session configuration registers and generate a new MAC value after modification	When opening a new session

Table 16: Conditional Self-Tests

The bypass capability of the cryptographic module is enabled on a frame-by-frame basis, depending on the cryptographic modes of the session open on the considered transponder.

- **No security**: full bypass.
- **Authentication only**: data used for authentication, but transmitted frame bypasses the ciphering/deciphering module.
- **Authentication & encryption**: data used for authentication and transmit the ciphered/deciphered data outputted by the AES-GCM algorithm.

For each transponder channel (sbt1, sbt2, kbt), two internal signals rx_crypto_mode and tx_crypto_mode are used to define the crypto mode of the ongoing session. The bypass mechanism integrity check is launched before each session opening (modification of the bypass mode) to verify that none of these crypto_mode signals have been corrupted since their last modification.

The bypass integrity test on these signals is based on the AES-GMAC authentication function, configured with a dedicated hardcoded key and IV values. At FPGA startup, the six crypto_mode signals are reset (no session open). The reset value of these signals are used as AAD in the AES-GMAC algorithm to initialize and store a

MAC value for each transponder. At each session opening, the value of the current rx_crypto_mode and tx_crypto_mode signals are re-injected in the AES-GMAC algorithm of RX and TX channels respectively to regenerate the MAC and compare them to the previously stored values (bit to bit comparison).

If one of the MAC comparison fails, the FPGA transit to the SELF-TEST ERROR state before reaching the degraded mode of operation, preventing further access to the ciphering, authentication and bypass functionalities.

If both MAC comparison succeed, the crypto_mode signals update is granted, and new MAC are generated and stored based on the crypto_mode signals of the new session configuration. The conditional bypass test is then launched.

The integrity check failures are reported by anomalies:

- 0xC2 - RX By-Pass Integrity Check Error
- 0xC3 - TX By-Pass Integrity Check Error

These anomalies are written in the the ‘Anomaly’ register that can be read by the processor through the local bus (show status service). In the absence of reported anomalies, the bypass integrity check is considered successful.

When a cryptographic session is opened, once the bypass mechanism integrity check and generation is done, a conditional bypass test is performed. A small frame, with 128bit AAD, 128bit data, 92bit IV and 256bits key (all FPGA hardcoded parameters) are sent to the frame processing function along with a frame source/destination signal corresponding to session being opened. Once processed, the output frame and MAC are verified to ensure that the frame has been processed according to the expected cryptographic mode (bypass, authentication only, authentication and ciphering). If a cryptographic session was already running, the previous test is repeated for the ongoing
session.

In case of failure of one of these test, the FPGA transit to the SELF-TEST ERROR state before reaching the degraded mode of operation, preventing further access to the ciphering, authentication and bypass functionalities.

Such failures are reported by anomalies:

- 0xC0 - RX Conditional By-Pass Test Error
- 0xC1 - TX Conditional By-Pass Test Error

These anomalies are written in the ‘Anomaly’ register that can be read by the processor through the local bus (show status service).

In the absence of reported anomalies, the conditional bypass self-test can be considered successful. The success of the conditional self-test is also indirectly reported by the sbt1/sbt2/kbt session status register indicating ‘Session Active’.

10.3 Periodic Self-Test Information

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
Bypass clear configuration	KAT (output in clear)	Bypass	N/A	N/A
Bypass Authentication only	KAT (MAC only)	Bypass	N/A	N/A
Bypass Authenticated encryption	KAT (MAC + data)	Bypass	N/A	N/A

Table 17: Pre-Operational Periodic Information

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
AES-GCM encryption	KAT	CAST	N/A	N/A
AES-GCM decryption	KAT	CAST	N/A	N/A
Conditional bypass test	KAT	Bypass	N/A	N/A
Bypass integrity test	MAC verification & generation	Bypass	N/A	N/A

Table 18: Conditional Periodic Information

<Text>

10.4 Error States

Name	Description	Conditions	Recovery Method	Indicator
INJ SELF-TEST ERROR	State reached if an error occurs during pre-operational self-tests in 'key loading' mode.	CRC error on key Error on AES-GCM KAT Error on pre-operational bypass self-test	Power OFF/Power ON	CTS signal remains high in case of failure.
SELF-TEST ERROR	State reached if an error occurs during pre-operational self-tests in 'nominal' mode, or during conditional self-tests.	CRC error on key Error on AES-GCM KAT Error on pre-	Power OFF/Power ON	Register value through "Show Status" Service

Name	Description	Conditions	Recovery Method	Indicator
		operational bypass self-test		

Table 19: Error States

After self-test operations in nominal mode, if the result is failed, the module enters in SELF-TEST ERROR state to set the status of the failure of the self-test. Then, it entries in degraded state. To exit this state, the equipment shall be powered-off and powered-on.

If the pre-operational self-test in key loading mode fails or if a failure occurs during the key loading process, the module reaches INJ SELF-TEST ERROR state. In this state, all the FPGA functions are disabled (all flip-flops wired to the FPGA output interfaces are under reset), including the key loading related services (CTS remains high). To resume normal operation, the module shall then be powered off and powered on.

10.5 Operator Initiation of Self-Tests

Pre-operational self-test and CAST can be run on demand by restarting the module.
Bypass mechanism integrity test and conditional bypass test can be run on demand by opening a session.

11 Life-Cycle Assurance

11.1 Installation, Initialization, and Startup Procedures

Once properly wired and powered on, no configuration is necessary for the module to operate and remain in the approved mode, as it is the only mode of operation of the module.

11.2 Administrator Guidance

To enable the cryptographic capabilities, operational traffic keys shall be loaded by the cryptographic officer through the DS-102 interface. After unplugging the key loader and rebooting the equipment, a user can configure the module to start communication sessions and access the cryptographic services.

11.3 Non-Administrator Guidance

On power-up, after successfully completing pre-operational and conditional self-tests, the CM transitions to an Approved mode operational state. In this state, the module awaits service requests from the operator.

[USER_MAN] provide non-administrator guidance.

11.4 Design and Rules

To organize and maintain the rights of each actor of the project, ClearCase is used for the different steps of development and verification. Palma is used for the documents organization. Each participant has specific rights if he is part of the project group. To get the exclusive right to modify any document in ClearCase or Palma, the participant has to check-out the document he is allowed to modify. In this case, any other participant cannot modify the document. To release the document and validate the modification, he has to check-in it. Each check-in operation is recorded in the database, and the internal version of the document is incremented for tracking.

Design elements (ClearCase)

ClearCase is used as configuration Management for the development of the project. Each file is versioned and labellized.

For each new delivery, a new version is identified (new label) in the configuration Management (ClearCase). All design elements receive the label using the command "cc_new_label", that tag all the files with the specified label.

The label takes this form: DELVRY_V'XX'_'YY'

- XX identifies an evolution of the specification document.
- YY identifies an intermediate release/delivery of the design.

When opening a configuration Clearcase view, the script allows to set and track the different tools versions. The needed tools and needed files version to work with are set.

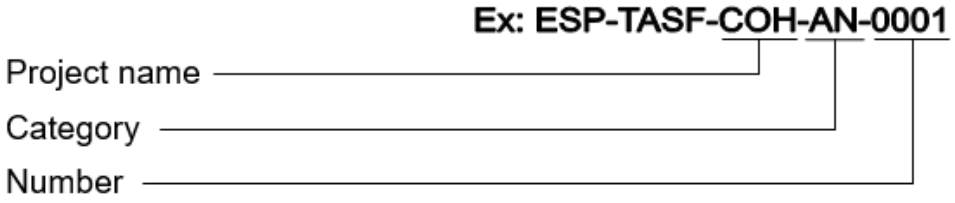
All files are identified by their ‘path+filename’.

Files are protected against modification when they are checked-in. Any checkout/modification/check-in creates a new version of the file. Different views are usable for different phase of design (RTL, verif, synthese, …).

Documents (PALMA)

All project documents are versioned on PALMA. Each document has a reserved space in PALMA. This space gives the document an identification depending category.

Category can be Design Document, Study and Analysis Report, User Manual, Verification Control Document…



The iterations of version are incremented by one: 001, 002, 003...

They are managed by the author of the document, using the "revise" function in Palma.

OPEN

PALMA creates internal iteration 001.x each time a Check-out/Check-in is done on the document.

Libraries/functions

The design files are sorted in libraries. These libraries refer to a specific function described in the specification. To identify the files associated to a function described in specification, the library has the same name as the function (Spacewire, micro, security…).

The design files represent a hierarchical architecture, so the main file name, for a library, is identified with the "_top" suffix. In this top file, other files (entities) are instantiated and usually named as their specific purpose (serializer, rx_management, sequencer_fsm …).

11.6 End of Life

At the end of mission, all SSP shall be zeroized. The procedure is described in §11.2 Zeroization. This procedure shall be realized by the CO only, at the end of mission. With this procedure, all the SSP are erased.

12 Mitigation of Other Attacks

N/A

13 References and Definitions

The Security Policy refers to the following documents.

Abbreviation	Document Name
[USER_MAN]	ESPRIT COM-HUB CRYPTOGRAPHIC MODULE USER MANUAL – v01

END OF DOCUMENT

OPEN