Datum Systems, Inc.

Stinger Encryption Module

FIPS 140-3 Non-Proprietary Security Policy

Document Version: v2.0.4
Date: 6/5/2026

Table of Contents

List of Tables

List of Figures

1 General

1.1 Overview

This document is the non-proprietary FIPS 140-3 Security Policy for Datum Systems' Stinger Encryption Module, hardware PN FIPS-24-001-1 version Rev C. It contains the security rules under which the module must operate and describes how this module meets the requirements as specified in FIPS PUB 140-3 (Federal Information Processing Standards Publication 140-3) for an overall Security Level 2 module.

1.2 Security Levels

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

Table 1: Security Levels

2 Cryptographic Module Specification

2.1 Description

Purpose and Use:

The Stinger Encryption Module (hereafter referred to as “the module”) is a cryptographic module designed for installation in Datum Systems modems, radios, and other communications equipment to provide secure Transmission Security (TRANSEC) for satellite, troposcatter and other wireless communication devices.

The module provides TRANSEC using approved AES-CBC-CS3 Datapath Encryption services for up to 4 independent transmit channels and AES-CBC-CS3 Datapath Decryption services for up to 32 independent receive channels.

Datapath Encryption services are provided for the radio transmitter, hereafter denoted by the prefix “Tx” or “tx” and Datapath Decryption services are provided for the radio receiver, hereafter denoted by the prefix “Rx” or “rx”. The plaintext and ciphertext data flow through the module is shown in Figure 1 below.

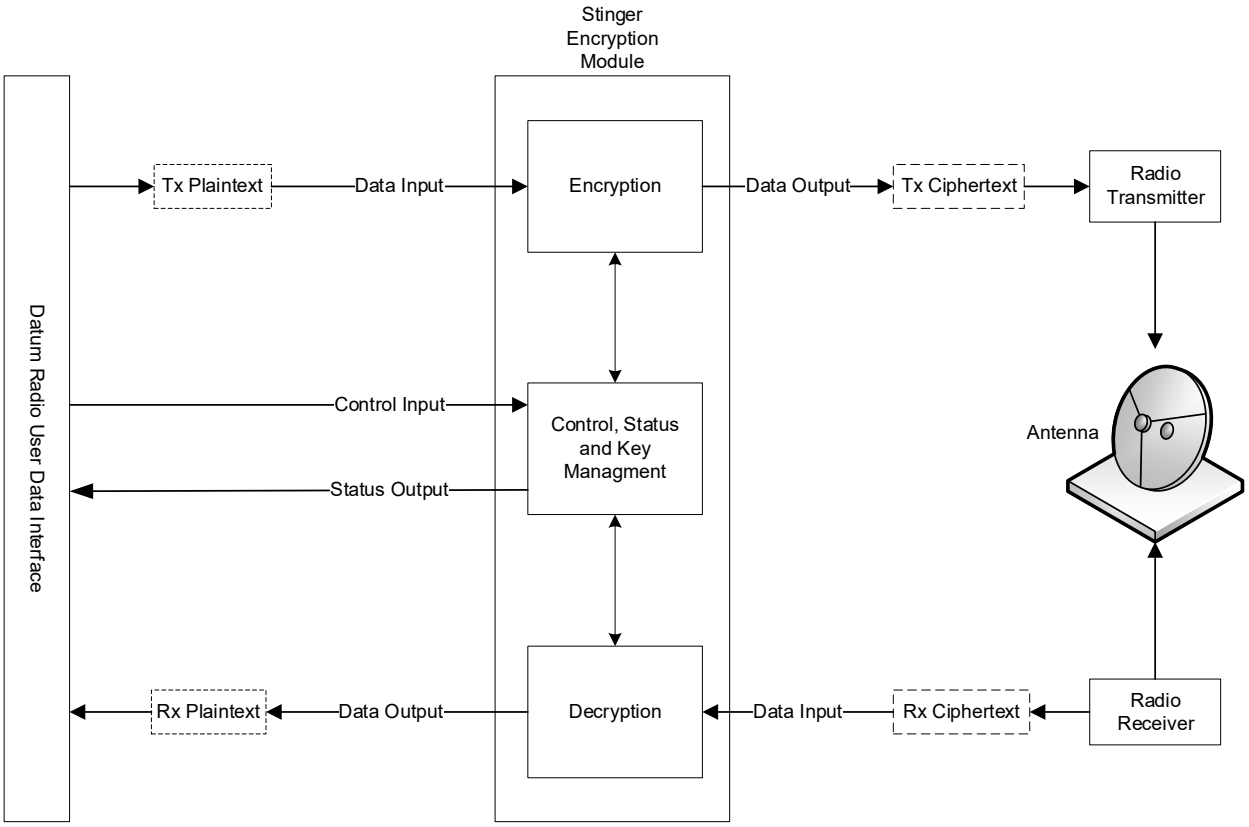

Figure 1: Module Data Flow

Datapath Encryption services are used exclusively by the radio transmitter. Plaintext channel data for radio transmission is input to the module from the radio's data interface. The corresponding channel ciphertext is output from the module and sent to the radio transmitter.

Datapath Decryption services are used exclusively by the radio receiver. Ciphertext channel data is received by the radio receiver and input to the module. The corresponding channel plaintext is output from the module and send to the radio's data interface.

Each channel can be configured by the Crypto Officer with a Key Derivation Key (KDK) that is used with a series of 16-bit Key Indices to derive a series of AES keys that are used for that channel's encryption or decryption. To prevent the key exhaustion, keys are rotated every 2^32 blocks for that channel. The key rotation is performed by incrementing the Key Index, deriving the next key in the rotation, and applying the key for encryption or decryption to the channel. In this document, the process of key rotation is hereafter referred to as a "key roll".

Encryption for a Tx channel is initiated by the Crypto Officer by loading a KDK into the module for the Tx channel using a serial command message. The Tx channel KDK must be zeroized first or the new KDK load will fail. Loading the new KDK implicitly loads the Tx channel key index to 1.

Decryption for an Rx channel is initiated by the Crypto Officer by loading a KDK into the module for the Rx channel using a serial command message. The Rx channel KDK must be zeroized first or the new KDK load will fail. Loading a new KDK implicitly loads the Rx channel key index to 1.

Datapath Encryption or Datapath Decryption services may be halted for a channel by zeroizing the KDK for that channel using a serial command message.

For the radio receiver, the operator must load the same channel KDK and key index used by the transmitter using a serial command message. The channels' key in the rotation corresponding the new key index is derived using an approved KDF using the key index and KDF. The channel KDF and key index must match for both the transmitter and receiver for the two radios in the link for successful channel decryption.

Module Type: Hardware

Module Embodiment: Multi-Chip Embedded

Module Characteristics:

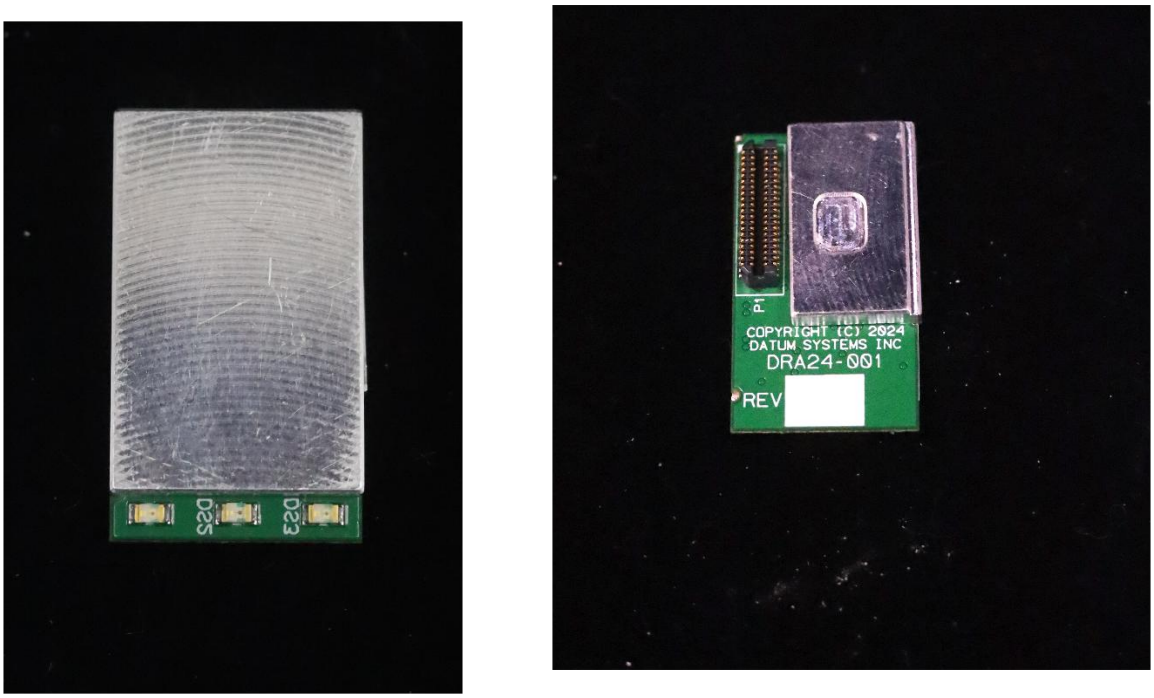

Figure 2: Stinger Cryptographic Module

The major active components on the module are listed below:

- Xilinx Artex7 Series FPGA that runs the module firmware. This is the primary cryptographic engine of the module. It also processes all serial commands and serial status replies.
- 1-megabit magnetroresistive RAM for persistent storage of all SSP keys and authentication credentials.
- Four 32-megabit flash memory devices used to store the FPGA configuration file.
- Microchip Technology, Inc Crypto-authentication device ATECC608C which is used to generate true random numbers using an SP800-90B NIST certified entropy source and DRGB. The entropy certificate is #E46. The DRGB was evaluated as part of the ATECC608C CAVP #A4280.

The module is controlled using a proprietary serial messaging protocol. Serial command messages are sent by the operator (Datum radio motherboard) through the control input interface to control the module, request cryptographic services, or request status. After the serial command is received and executed by the module, a serial message response is sent back to the operator through the status output interface. Every serial command message has a corresponding serial response message. There is also a Fault status message used for logging purposes that is sent to the status output interface anytime the module fails a self-test and enters the error state. This is the only serial output message that is sent that is not in response to a serial command message (see section 10 Self-Tests).

Cryptographic Boundary:

The cryptographic boundary of the module is the physical perimeter of the module assembly as shown in Figure 2, and encompasses the entire module PCB and all hardware and firmware components contained therein. No components on the module are excluded from the cryptographic boundary. All SSPs used by the module reside inside the boundary.

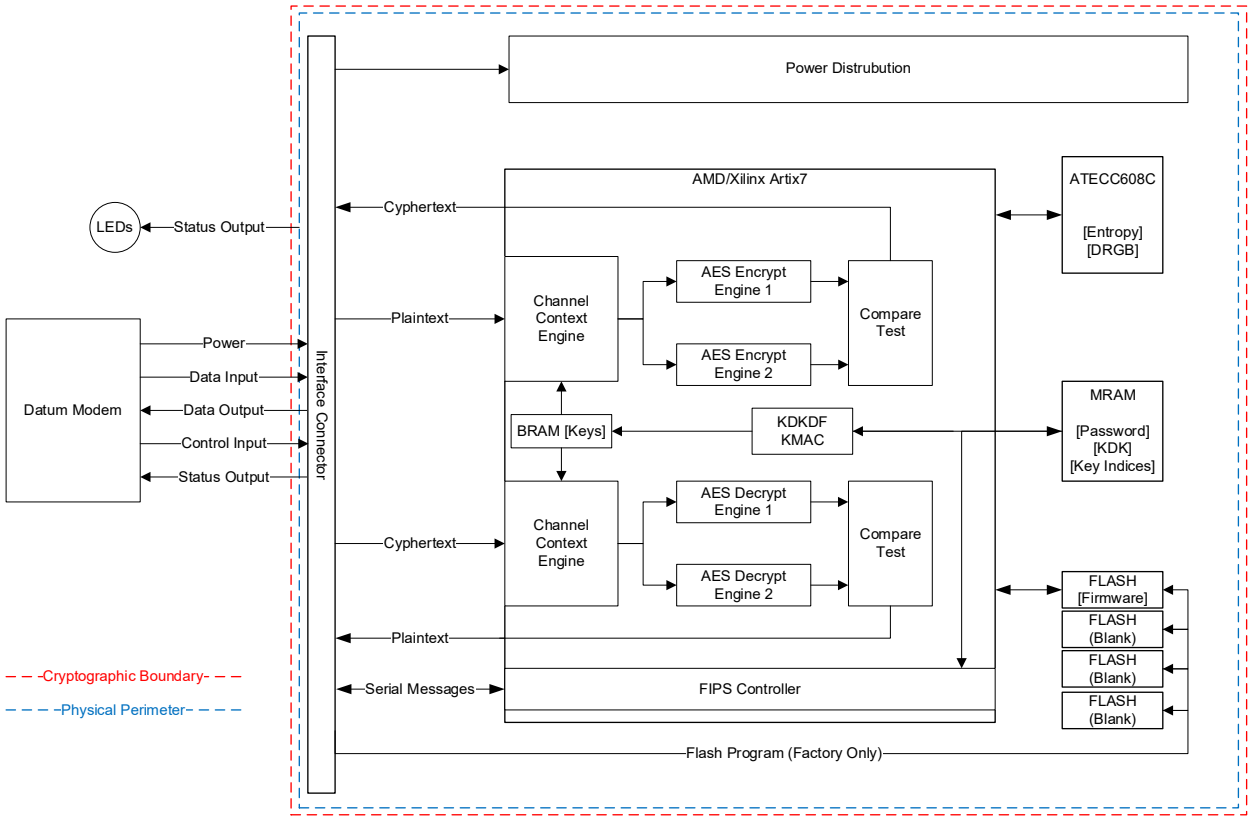

Figure 3: Block Diagram

Multi-channel plaintext Tx data with channel identifiers enter the module through the Data Input port and are processed by a context channel engine to assign the correct encryption context (key, previous ciphertext block, etc.) to two instances of identical AES-CBC-CS3 encryption engines that are used to encrypt the data. The encrypted ciphertext blocks from both engines are then sent to a continuous comparison test that compares all bits of both blocks. If the blocks are identical then the cyphertext is sent along with channel identification to the Data Output interface. If the ciphertext blocks do not match, the module enters the HW Error state where all cryptographic services are halted.

Tx plaintext data is input to the module as packets that consist of a two-byte header and plaintext byte data for encryption. The first header byte is the Tx channel for the input plaintext data. The second header byte contains two bit flags to mark the start of a string (Start Flag) and to mark when to begin using a new key for a key roll (NextKey Flag). The remaining 6 bits are passed through the module to the ciphertext output. The plaintext byte data can be any number of bytes >= 64 bytes.

Tx ciphertext data is output from the module as packets that consist of a two-byte header and the encrypted ciphertext data. The first byte of the header is the Tx channel for the output ciphertext data. The second byte of the header contains a bit flag to mark the start of the encrypted string (Start Flag) and a bit flag to indicate that the encrypted data is valid (Valid Flag). The remaining 6 bits are the passed through from the plaintext input.

The module maintains a block count for each encryption (Tx) channel which increments on every new encrypted ciphertext block. When the channel block count approaches 2^32 blocks

a key roll request flag bit is set in the get_status message. The operator is responsible for polling the module through the Control Input and Status Output interface using the get_status serial command message and setting the NextKey flag in the Tx plaintext packet header for the channel input data through the Data Input interface when a Tx key roll is requested. If the operator does not properly set the NextKey flag within 6 seconds of the request, the module will perform the key roll without a NextKey flag.

When a key roll occurs, the key index is incremented and a new key is generated using an approved KDF using that channels' new key index and KDK. The encryption for that channel begins using the new key when the NextKey flag is present in the input data for that channel. When a channels' key index reaches 65535, encryption for that channel ceases, the module outputs all zero data for the channel, and an alarm is indicated for the channel. The operator must then zeroize the KDK for the channel and load a new KDK into the module for that channel to continue the Datapath Encryption service for the channel.

In a similar manner, multi-channel ciphertext Rx data with channel identifiers enter the module through the Data Input port and are processed by a context channel engine to assign the correct decryption context (key, previous block, etc.) to two instances of identical AES-CBC-CS3 decryption engines that are used to decrypt the data. The decrypted plaintext blocks from both engines are then sent to a continuous comparison module that compares all bits of both plaintext blocks. If the blocks are identical then the plaintext is sent along with channel identification to the Data Output interface. If the plaintext blocks do not match the module goes to the HW Error state where all cryptographic services are halted.

Rx cyphertext data input to the module as packets that consist of a two-byte header and ciphertext byte data for decryption. The first byte of the header is the Rx channel for the input ciphertext data. The second byte of the header contains a bit flag to mark the start of a string (Start Flag) and the bit flag to mark when to begin using a new key for a key roll (NextKey Flag). The remaining 6 bits are passed through the module to the plaintext output. The ciphertext byte data can be any number of bytes >= 64 bytes.

Rx plaintext data is output from the module as packets that consist of a two-byte header and the decrypted plaintext byte data. The first byte of the header is the Rx channel for the output plaintext data. The second byte of the header contains a bit flag to mark the start of the decrypted string (Start Flag) and a bit flag to indicate that the decrypted data is valid (Valid Flag). The remaining 6 bits are passed through from the ciphertext input.

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
FIPS24-001-1	Rev C	0x251109102609	Xilinx Artix7	

Table 2: Tested Module Identification – Hardware

The module hardware Model Number and Hardware version are retrieved from the module using the get_name serial command message. The get_name serial response message has the formatting as shown in the table below.

Bit(s)	Value	Type	Description
7-0	0x77	Hex	Serial Response Header for “get_name” Command
15-8	0x00	Hex	Always Zero
111-16	“FIPS24-001-1”	ASCII	Model/Part Number
119-112	“ “	ASCII	ASCII Space Character
151-120	“Rev C”	ASCII	Hardware Version
279-152	[Ignore]	N/A	Ignore these bits

Table 3: get_name Serial Response Message Bit Map

The module firmware version is retrieved by the get_id serial command message. The get_id serial response message has formatting as shown in the table below.

Bit(s)	Value	Type	Description
7-0	0x76	Hex	Serial Response Header for “get_id” Command
15-8	0x00	Hex	Always Zero
87-16	Varies	Dec	Module Serial Number
145-88	Varies	Dec	FPGA Serial Number
191-146	0	Dec	All zeros ‘0’
239-192	0x251109102609	Hex	Firmware Version
247-240	0x10	Hex	Build Code
271-248	0x300082	Hex	PCB ID
279-272	[Ignore]	N/A	Ignore these bits

Table 4: get_id Serial Response Message Bit Map

2.3 Excluded Components

No components of the module are excluded from the cryptographic boundary.

2.4 Modes of Operation

Modes List and Description:

Mode Name	Description	Type	Status Indicator
Approved Mode	TRANSEC	Approved	Operational Pin='1'. Operational LED=On

Table 5: Modes List and Description

The module has a single mode of operation. This is the approved mode and is entered by resetting the module after power-up. If a failure occurs the module enters a HW Error state where all cryptographic services are halted and which cannot be exited except by resetting the module, with or without power-cycling.

2.5 Algorithms

Approved Algorithms:

The table below lists the approved security functions (or cryptographic algorithms) of the module, including specific properties.

Algorithm	CAVP Cert	Properties	Reference
AES-CBC-CS3	A7740	Direction - decrypt, encrypt Key Length - 128, 192, 256	SP 800-38A
AES-ECB	A4280	Direction - Decrypt, Encrypt Key Length - 128	SP 800-38A
Counter DRBG	A4280	Prediction Resistance - No Mode - AES-128 Derivation Function Enabled - Yes	SP 800-90A Rev. 1
KDF KMAC Sp800-108r1	A7740	Derived Key Length - Derived Key Length: 128, Derived Key Length: 192, Derived Key Length: 256	SP 800-108 Rev. 1
KMAC-256	A7740	Message Length - Message Length: 16 Key Data Length - Key Data Length: 128, 192, 256	SP 800-185
SHA3-256	A7740	Message Length - Message Length: 128-256 Increment 64	FIPS 202

Table 6: Approved Algorithms

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
Encryption	BC-UnAuthEncrypt	Block Cipher Symmetric Encryption Non-Authenticated	Key Size:128, 192, 256	AES-CBC-CS3: (A7740)

Name	Type	Description	Properties	Algorithms
Decryption	BC-UnAuthDecrypt	Block Cipher Symmetric Decryption Non-Authenticated	Key Size:128, 192, 256	AES-CBC-CS3: (A7740)
Key Roll	KBKDF	Derive a new key when a channel block count approaches 2^32 blocks. Derive first key when a new key is loaded	Key Size:128, 192, 256	KDF KMAC Sp800-108r1: (A7740) KMAC-256: (A7740)
KDK Hash	SHA	Output KDK Hash for entry verification		SHA3-256: (A7740)
DRBG	DRBG	Random Number Generation	Derivation Function: Enabled	Counter DRBG: (A4280) AES-ECB: (A4280)
Authentication	UNK	Operator credential authentication		

Table 7: Security Function Implementations

2.7 Algorithm Specific Information

N/A

2.8 RBG and Entropy

Cert Number	Vendor Name
E46	Microchip Technology Inc

Table 8: Entropy Certificates

Name	Type	Operational Environment	Sample Size	Entropy per Sample	Conditioning Component
ECC608 NRBG Entropy Source	Physical	ATECC608C	1	0.5071	

Table 9: Entropy Sources

The ATECC608C entropy source provides full entropy output to initialize the approved DRBG.

2.9 Key Generation

The DRBG is used to generate AES IVs for encryption and decryption, as well as random strings for the tx_trn service.

The module supports an approved SP 800-108r1 key derivation function using KMAC.

When activating the Datapath Encryption Service or Datapath Decryption Service, the CO operator loads a KDK into the module for one Tx or Rx channel. When the KDK is loaded, the DRBG 256-bit output is truncated to 128 bits and is loaded at the initialization vector (IV) for the channel Once the KDK is loaded and activated using the set_tx_key or set_rx_key serial command message, the KDK and Key Index for the channel are used with an approved KDF to derive the key that is used for encryption of a Tx channel or decryption of an Rx channel. When a Tx key roll occurs (which increments the Tx channel Key Index) or a new Rx channel Key Index is loaded into the module, a new key is derived using the approved KDF. See Section 2.1 Description for details.

2.10 Key Establishment

The module does not support any automated SSP establishment method.

2.11 Industry Protocols

The module does not support interoperation with any established Industry Protocols.

3 Cryptographic Module Interfaces

3.1 Ports and Interfaces

The module's design separates the physical ports into five logically distinct and isolated categories:

- Data Input
- Data Output
- Control Input
- Status Output
- Power

The module plugs directly into the motherboard of a Datum radio through an 80-pin high-density (HD) connector referred to hereafter as the P1 Port. There are also three LEDs that are used as part of the Status Output interface to indicate module status:

- DS1: Init Error LED
- DS2: HW Error LED
- DS3: Operational LED

The table below provides a mapping of the physical interface pins and ports to the logical interfaces as well of a description of the pin/port functions.

Physical Port	Logical Interface(s)	Data That Passes
A2, A4, A5, A6, A7 (TxPText)	Data Input	Tx Plaintext Data
B2, B4, B5, B6, B7 (RxCText)	Data Input	Rx Encrypted Data
C2, C4, C5, C6, C7 (TxCText)	Data Output	Tx Encrypted Data
D2, D4, D5, D6, D7 (RxPTextD)	Data Output	Rx Plaintext Data
A17 (CpuWr)	Control Input	Serial Message
A16 (KeyWr)	Data Input	Key Entry
B17 (nPgmCfg)	Control Input	Module Reset
C19 (MasterClk)	Control Input	Module Clock Ref
A18, B18 (PgmSel)	Control Input	Mode Select
A19, A20, B20, D17 (PgmCtl)	Control Input	Flash Programming
B16 (Done)	Status Output	Init Error
C16 (Error)	Status Output	HW Error
D16 (Operational)	Status Output	Operational
D18 (GND)	Status Output	Module Installed
C17 (CpuRd)	Status Output	Serial Message
DS1 (Init Error LED)	Status Output	Initialization Error Indicator
DS2 (HW Error LED)	Status Output	Hardware Error Indicator
DS3 (Operational LED)	Status Output	Operating in Approved Mode
A1, B1, C1, D1, A3, B3, C3, D3, A8, B8, C8, D8, A11, B11, C11, D11, A13, B13, C13, D13, A15, B15, C15, D15, C18, B19, D19, C20, D20 (GND)	Power	GND
A9, B9, C9, D9, A10, B10, C10, D10 (Power)	Power	+1.0V DC
A12, B12, C12, D12 (Power)	Power	+2.5V DC
A14, B14, C14, D14 (Power)	Power	+3.3V DC

Table 10: Ports and Interfaces

Notes:

- The module does not support a Control Output Interface.
- The Flash Programming pins are only used during factory programming. After factory programming and programming verification, all the Flash IC devices are rendered read only permanently using flash device OTP registers.
- The Flash IC devices data pins are connected internally only to the FPGA, so no readback is possible via any of the module's interfaces.

- The Module Select inputs must be set to P1-B18=GND and P1-B19=GND during module reset. Any other input results in module failure and an Init Error indication.
- The HMI for the module consists of:
  1. P1 is a high-density electrical connector with pin groups separated into distinct and isolated logical interfaces.
     a. Data Input Interface
     b. Data Output Interface
     c. Control Input Interface
     d. Status Output Interface
     e. Power Interface
  2. 3 LEDs are part of the Status Output Interface and provide continuous status of the module.

3.2 Control Interface Not Inhibited

When there is any failure in power up, self-test, conditional self-test, periodic self-test or continuous self-test, the module enters the HW Error state. If any of the supply voltages an/or the junction temperature fall outside the acceptable range, the module enters the EFP Error state. When in either the HW or EFP error state, the module sends an error log out the Status Output serial interface, and the Control Input interface is inhibited with the exception of the reset control, Pin B17. Also, when in either error state, the Status Output interface is inhibited with the exception of the LED indicators, the HW Error pin, and the Init Error pin. In either error state both the Data Input and Data Output interfaces are disabled. The only way to exit the HW Error state is to reset the module. In the Init Error state, the FPGA is not configured and no cryptographic services nor input output is possible. The only way to exit the Init Error state is to reset the module.

4 Roles, Services, and Authentication

4.1 Authentication Methods

The authentication mechanism is designed to meet the following objectives:

- For each attempt, the probability shall be less than one in 1,000,000 that a random attempt will succeed or a false acceptance will occur.
- For multiple attempts during a one-minute period, the probability shall be less than one in 100,000 that a random attempt will succeed or a false acceptance will occur.

Method Name	Description	Security Mechanism	Strength Each Attempt	Strength per Minute
Password	Password authentication mechanism	Memorized Secret	2.93E^-39 (module enforces 16-byte identifier)	1.19E^-31

Table 11: Authentication Methods

The module supports only role-based authentication. Unique 128-bit identification passwords are sent from the operator (Datum radio motherboard) with each message that requires authentication to assume the role performing the service. Authentication is required for each message as no authentication information nor role state is preserved between each service request.

Before the module can perform any security functions or services, User and CO credentials must be configured. User and CO credentials cannot be set unless all SSPs are zeroized. The module is zeroized before leaving the factory, or the module can be zeroized using the zero_all serial command message. Once zeroized, the User and CO credentials can be loaded into the module by issuing the set_user and set_co serial command messages.

The unique User and CO identifiers are both 128-bits long. The probability of guessing the identifier is 1 in 2^128 or 2.93E-39. The fixed bit rate of the serial control input interface is 125 megabits/second, and serial commands that require authentication require 185 bits. Thus, rounding up to the nearest 1000, 676,000 validate requests can be passed to the module per second assuming no processing delay, overhead, or serial response message time. This results in 40,560,000 attempts per minute, and a 1.19E-31 strength per minute.

4.2 Roles

The module supports two distinct operator roles. The two roles supported by the module are Crypto Officer and User. The module does not support a Maintenance role. Authentication is required for all Crypto Officer services. Some User services require authentication and some User services do not require authentication. The module supports only a single operator, the Datum radio motherboard, that assumes one of the two roles for each serial command message.

Name	Type	Operator Type	Authentication Methods
Crypto Officer	Role	Crypto Officer	Password
User	Role	User	Password

Table 12: Roles

Each service request is made through a serial command message on the Control Input interface, and during the serial command message processing the module cannot accept any further commands until the current command is completed and serial response message sent out the Status Output Interface. Thus, the module cannot support concurrent service requests nor concurrent operators.

4.3 Approved Services

The module provides services to the operator who assumes an available role with each command message. The module only supports approved services and always operates in approved mode. Use of approved services is indicated by toggling of hardware lines on the modules Control Input and Status Output interfaces conforming to a proprietary serial messaging protocol.

The module does not support any bypass modes.

The following table lists the approved services available on the HMI that utilize approved security functions.

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
set_user	Configure User Password	Serial Message Response	Command and New User Password	OK or ERR	None	Unauthenticated - User Password: W
set_co	Configure CO Password	Serial Message Response	Command and New CO Password	OK or ERR	None	Unauthenticated - CO Password: W
zero_all	Zeroize All SSPs (KDKs, Key Indices, User Password, and CO Password)	Serial Message Response	Command	OK or ERR	None	Unauthenticated - CO Password: Z - User Password: Z - KDK: Z - KDK Hash: Z - Key Index: Z - AES Key: Z - AES IV: Z
self_test	Perform Self-Test	Serial Message Response	Command	OK or ERR	None	Unauthenticated
get_id	Get Module Firmware Version	Serial Message Response	Command	Firmware Identifier	None	Unauthenticated
get_name	Get Module Part Number and Hardware Version	Serial Message Response	Command	Hardware Identifiers	None	Unauthenticated
zero_tx	Zeroize Encryption Channel KDK and Key Index	Serial Message Response	Command & Channel	OK or ERR	None	User - KDK: Z - AES Key: Z - Key Index: Z - User Password: E
zero_rx	Zeroize Decryption Channel	Serial Message Response	Command & Channel	OK or ERR	None	User - KDK: Z - AES Key: Z - Key Index:

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
	KDK and Key Index					Z - User Password: E
set_rx_index	Set One Decryption Channel's Key Index, Derive and Load new Key	Serial Message Response	Command, Channel & new Key Index	OK or ERR	Key Roll Authentication	User - AES Key: G,E - Key Index: W - User Password: E
get_status	Get Module Status	Serial Message Response	Command	Status Code	Authentication	User - User Password: E
get_tx_hash	Get One Encryption Channel KDK Hash	Serial Message Response	Command & Channel	KDK Hash or ERR	KDK Hash Authentication	User - KDK Hash: G,R - User Password: E
get_tx_index	Get One Encryption Channel Key Index	Serial Message Response	Command & Channel	Key Index or ERR	Authentication	User - Key Index: R - User Password: E
get_rx_hash	Get One Decryption Channel KDK Hash	Serial Message Response	Command	KDK Hash or ERR	KDK Hash Authentication	User - KDK Hash: G,R
get_rx_index	Get One Decryption Channel Key Index	Serial Message Response	Command & Channel	Key Index or ERR	Authentication	User - Key Index: R - User Password: E
get_trn	Get 256-bit Random Number	Serial Message Response	Command	OK or ERR	DRBG Authentication	User - User Password: E - Entropy Input: G,E - DRBG Seed: G,E - DRBG V: G,E - DRBG Key: G,E
co_valid	Validate CO Password	Serial Message Response	Command/ CO Password	OK or ERR	Authentication	Crypto Officer - CO Password: E - User Password: E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
set_tx_key	Load One Encryption Channel KDK	Serial Message Response	Command & KDK	OK or ERR	Key Roll Authentication	Crypto Officer - KDK: W - AES Key: G - Key Index: W - CO Password: E
set_tx_roll	Increment One Encryption Channel Key Index, Derive and Load new Key	Serial Message Response	Command	OK or ERR	Key Roll Authentication	Crypto Officer - AES Key: G - Key Index: W - CO Password: E
set_rx_key	Load One Decryption Channel KDK	Serial Message Response	Command & KDK	OK or ERR	Key Roll Authentication	Crypto Officer - KDK: W - AES Key: G - Key Index: W - CO Password: E
Datapath Encryption	Encrypt Tx Channel plaintext. Automatically performed when KDK and Key Index are provisioned for a channel using serial command message, and Tx Plaintext Data Packet loaded into module on Data Input interface.	Bits 126-119 of the get_status serial command response.	Tx Channel plaintext data packets	Tx Channel ciphertext data packets	Encryption	Crypto Officer - AES Key: E
Datapath Decryption	Decrypt Rx Channel - Automatica	Bits 189-126 of the get_status	Rx Channel plaintext	Rx Channel	Decryption	Crypto Officer - AES Key: E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
	lly performed when KDK and Key Index are provisioned for a channel using serial command message, and Rx Ciphertext Data Packet is loaded into module on Data Input interface	serial command response.	data packets	ciphertext data packets		
Automatic Tx Key Roll	Tx Channel Key Roll - Automatically performed when number of encrypted blocks for a channel is 2^32 blocks to prevent key exhaustion.	Bits 118-115 of the get_status serial command response. get_tx_index serial response message.	None	New AES Key	Key Roll	Unauthenticated - AES Key: G,W - Key Index: W

Table 13: Approved Services

Approved Services Table Notes:

- CO=Crypto Officer Role, US=User Role requiring authentication.
- The Sensitive Security Parameters (SSPs) list tin the above table are defined with the following access notations:
    - (G)enerate: The module generates or derives the SSP.
    - (R)ead: The SSP is read from the module (e.g., the SSP is output).
    - (W)rite: The SSP is updated, imported, or written to the module.
    - (E)xecute: The module uses the SSP in performing a cryptographic or authentication operation.
    - (Z)eroize: The module zeroizes the SSP.

- OK: Indicates the OK serial response message on the CpuRd pin of the Status Output interface.
- ERR: Indicates the ERR serial response message on the CpuRd pin of the Status Output Interface.
- set_user: Can only be used when the module is zeroized, otherwise returns ERR.
- set_co: Can only be used when the module is zeroized and the User password has been loaded with the set_user command, otherwise returns ERR.
- set_tx_key: Channel KDK must be zeroized first, otherwise returns ERR.
- set_rx_key: Channel KDK must be zeroized first, otherwise returns ERR.
- All CO serial message commands require a two-step process:
    1. Send the CO 128-bit credential into the module through the KeyWr pin. For the set_tx_key and the set_rx_key the CO 128-bit credential and the KDK is sent on KeyWr.
    2. Send the CO command into the module through the CpuWr pin (requires the User credential).

  These steps must be done exactly as listed above. If an incorrect CO credential is sent is step 1, step 2 will fail with response ERR. If step 2 is performed without step 1, the response will be ERR. If step 2 has an incorrect User credential, the module is zeroized. If any other command message besides a CO command is sent for step 2, the response will be ERR.

The module supports self-initiated cryptographic output functionality when using the Datapath Encryption and/or Datapath Decryption service. Before enabling these services, the CO must configure and perform two independent actions in order to activate the services:

1. The CO must authenticate and load a KDK record into the module on the KeyWr pin.
2. The CO must activate the Datapath Encryption or Datapath Decryption service by sending the set_tx_key or set_rx_key serial message command on the CpuWr pin.

Note if step 2 is not performed, the KDK record will be erased on any other serial command message.

4.4 Non-Approved Services

N/A for this module.

The module does not support any Non-Approved Services.

4.5 External Software/Firmware Loaded

The module does not support External Software/Firmware Load.

4.6 Cryptographic Output Actions and Status

For the purposes of verification, the CO password can be verified using the co_valid serial command. For key entry verification, the Key Hash PSP for each Tx and Rx KDK can be retrieved from the module using the get_tx_hash and get_rx_hash serial command messages. For key roll status, the current Key Indices can be retrieved using the get_tx_index and get_rx_index serial command messages. There is also a convenience command used by the

operator in the User role to get a random number from the approved random number generator via the get_trn serial command message. The random number generated and retrieved through this call is not used for any security or non-security purpose within the module. Note that all CO role serial commands required both the CO password on the KeyWr pin and the User password on the CpuWr pin as described in the notes to the Approved Services table.

5 Software/Firmware Security

5.1 Integrity Techniques

The firmware for the module consists of a single FPGA bitstream file. The FPGA contains hardware that automatically performs an integrity check of this file during FPGA configuration using a CRC32 EDC during each power-up/reset sequence.

During configuration, a hardware module within the FPGA computes a CRC32 for each data block in the configuration bitstream file. If the CRC32 does not match the CRC32 value for that block (also included in the bitstream), then the FPGA will not configure properly and the module will enter the Init Error State as indicated by the Init Error pin (Pin B16) on the Status Output interface.

5.2 Initiate on Demand

The operator can initiate this CRC32 EDC check by resetting the module.

6 Operational Environment

6.1 Operational Environment Type and Requirements

Type of Operational Environment: Non-Modifiable

The module operates in a non-modifiable operational environment. The module's firmware is programmed at the factory and cannot be changed after manufacturing. The module's firmware version 0x251109102609 runs on an AMD/Xilinx Artix7 FPGA. The FPGA protects memory and process space from unauthorized access. The CRC32 EDC check protects against errors in the bitstream file store in the flash memory which is located within the cryptographic boundary.

7 Physical Security

7.1 Mechanisms and Actions Required

The module is installed into a Datum radio at the factory and is not accessible by the operator. There are no operator inspection or test intervals or recommendations.

Mechanism	Inspection Frequency	Inspection Guidance
Tamper-proof encapsulating material	Inspected at factory. Module is completely enclosed by the Datum radio chassis, and is not accessible by the operator of the radio. If the operator suspects the radio has been tampered with, the operator should immediately take the radio out of service and contact Datum Systems.	Factory only inspection: Look for damage to the metal covers on the module's PCB. If the metal covers are missing or damaged, look for any damage to the epoxy coating. If the metal covers are missing or damaged, or the epoxy is damaged and the module is still functioning, zeroize the module and securely destroy the module.

Table 14: Mechanisms and Actions Required

8 Non-Invasive Security

N/A for this module.

9 Sensitive Security Parameters Management

9.1 Storage Areas

Storage Area Name	Description	Persistence Type
MRAM	Magnetroresistive RAM	Static
FPGA BRAM	FPGA Block RAM/Registers	Dynamic
ATECC608C	Security IC	Dynamic

Table 15: Storage Areas

9.2 SSP Input-Output Methods

Name	From	To	Format Type	Distribution Type	Entry Type	SFI or Algorithm
SSP Input	Datum radio motherboard	MRAM	Plaintext	Manual	Electronic	
SSP Output	FPGA BRAM	Datum radio motherboard	Plaintext	Manual	Electronic	

Table 16: SSP Input-Output Methods

Notes: Distribution and Entry Type per [IG 9.5.A].

Inputs:

- KDK (CSP): KDKs are loaded to the module from the Datum radio motherboard using the set_tx_key or set_rx_key serial command message for each channel in use.
- Key Index (PSP): Tx (encryption) and Rx (decryption) key indices are loaded into the module using the set_tx_roll or set_rx_index serial command for each channel in use.

Outputs

- KDK Hash (PSP): A hash for a stored KDK is output from the module to the Datum radio motherboard using the get_tx_hash or get_rx_hash serial command message.
- Key Index (PSP): The Tx (encryption) and Rx (decryption) key indices for any channel are output from the module to the Datum radio motherboard using the get_tx_index and get_rx_index serial command messages.

Note: The channel KDK and Key Index are used to derive the key used for that channel's encryption or decryption.

9.3 SSP Zeroization Methods

Zeroization Method	Description	Rationale	Operator Initiation
Serial Command	The zeroization is performed by the module overwriting zeroes to the memory location occupied by the SSP. Successful Zeroization is indicated by an OK response from the serial command.	The operator (Datum radio), interacting with the module, is responsible for calling the appropriate zeroization serial command messages listed in the above table to zeroize the SSPs. The completion of a zeroization serial command with an OK serial response will indicate that a zeroization procedure succeeded.	By invocation through serial command message.
Incorrect User Credentials	Any User serial message command received with incorrect User credentials will zeroize the module, with the exception of the get_status serial command message.	Incorrect user credentials are not allowed, with the exception of the get_status serial command message. Zeroization of the module is indicated by the ERR response message.	Send any serial command message that requires user credentials with the incorrect user credentials.

Table 17: SSP Zeroization Methods

There are three zeroization serial commands available: zero_tx, zero_rx and zero_all_keys.

- zero_tx: Zeroizes one encryption channel KDK, Key, and Key Index.
- zero_rx: Zeroizes one decryption channel KDK, Key, and Key Index.
- zero_all_keys: Zeroizes all encryption and decryption channel KDKs, Keys, Key Indices, User password and CO Password.

9.4 SSPs

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
User Password	User Password	128 - 128	Memorized Secret - CSP			Authentication
CO Password	Crypto Officer Password	128 - 128	Memorized Secret - CSP			Authentication
KDK	AES key used as input for KBKDF KMAC	128, 192, 256 - 128, 192, 256	Symmetric Key - CSP			KDF KMAC Sp800-108r1 (A7740)
AES Key	AES key used for Encryption and Decryption	128, 192, 256 - 128, 192, 256	Symmetric Key - CSP	KDF KMAC Sp800-108r1 (A7740)		Encryption Decryption
AES IV	DRBG output truncated to 128 bits and used internally as initialization vector	128 - 128	Random Bits - PSP	DRBG		Encryption Decryption
KDK Hash	Hash Used for KDK Entry Verification	256 - 256	Hash - PSP	KDK Hash SHA3-256 (A7740)		KDK Hash
Key Index	Index used by KDF to Generate AES Key used for Encryption/Decryption	16 -	KDF Input - PSP	Key Roll		Key Roll
Entropy Input	Entropy material for DRBG	256 - 256	Entropy Input - CSP	ATECC608C		DRBG
DRBG Seed	Seeding material for DRBG	384 - 256	Seed - CSP	DRBG		DRBG
DRBG V	DRBG internal state value	256 - 256	Internal State Value - CSP	DRBG		DRBG
DRBG Key	DRBG internal state value	128 - 128	Internal State Value - CSP	DRBG		DRBG

Table 18: SSP Table 1

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
User Password	SSP Input	MRAM:Plaintext		Serial Command Incorrect User Credentials	
CO Password	SSP Input	MRAM:Plaintext		Serial Command Incorrect User Credentials	
KDK	SSP Input	MRAM:Plaintext		Serial Command Incorrect User Credentials	AES Key:Derives
AES Key		FPGA BRAM:Plaintext		Serial Command Incorrect User Credentials	KDK:Derived From
AES IV		FPGA BRAM:Plaintext	Ephemeral	Serial Command Incorrect User Credentials	AES Key:Used With
KDK Hash	SSP Output	FPGA BRAM:Plaintext		Serial Command Incorrect User Credentials	KDK:Paired With
Key Index	SSP Input SSP Output	MRAM:Plaintext		Serial Command Incorrect User Credentials	AES Key:Derives
Entropy Input		ATECC608C:Plaintext	Ephemeral	Serial Command Incorrect User Credentials	DRBG Seed:Derives
DRBG Seed		ATECC608C:Plaintext	Ephemeral	Serial Command Incorrect User Credentials	Entropy Input:Derived From DRBG V Value:Derives DRBG Key Value:Derives
DRBG V		ATECC608C:Plaintext	Ephemeral	Serial Command Incorrect User Credentials	DRBG Seed:Derived From
DRBG Key		ATECC608C:Plaintext	Ephemeral	Serial Command Incorrect User Credentials	DRBG Seed:Derived From

Table 19: SSP Table 2

10 Self-Tests

10.1 Pre-Operational Self-Tests

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details
CRC32	32 bits	EDC	SW/FW Integrity	LED	EDC for FPGA configuration per [IG 10.3.F]
Entropy Start-up Health Tests	False positive rates of 2^-20 (APT) and 2^-50 (RCT)	APT and RCT	Critical Function	LED	Sp 800-90B power-up health tests

Table 20: Pre-Operational Self-Tests

After reset, the module enters the Initialization state where the FPGA begins reading the bitstream file from Flash memory for configuration. The FPGA then enters the Integrity Check state, where the FPGA reads each block of the bitstream file from Flash memory, computes a CRC32 value for that block and compares the computed CRC32 value to a reference CRC32 value stored in the bitstream file for that block. If any comparison fails, the FPGA will not configure and the module goes to the Init Error state. If all the blocks have proper CRC32 value, the FPGA configuration completes and the module enters the Self-Test State.

10.2 Conditional Self-Tests

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
SHA3	256	KAT	CAST	Serial Response Message, HW Error LED=Off, HW Error pin='0'	KDK Hash	Reset, Periodic
AES Encrypt 1	128, 192, 256	Comparison	CAST	Serial Response Message, HW Error LED=Off, HW Error pin='0'	Encryption for Encryption Engine Instance 1	Reset, self_test serial command, Periodic
AES Encrypt 2	128, 192, 256	Comparison	CAST	Serial Response Message, HW Error LED=Off,	Encryption for Encryption Engine Instance 2	Reset, self_test serial command, Periodic

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
				HW Error pin='0'		
AES Decrypt 1	128, 192, 256	Comparison	CAST	Serial Response Message, HW Error LED=Off, HW Error pin='0'	Decryption for Decryption Engine Instance 1	Reset, self_test serial command, Periodic
AES Decrypt 2	128, 192, 256	Comparison	CAST	Serial Response Message, HW Error LED=Off, HW Error pin='0'	Decryption for Decryption Engine Instance 2	Reset, self_test serial command, Periodic
KBKDF KMAC	256	KAT	CAST	Serial Response Message, HW Error LED=Off, HW Error pin='0'	Key Roll Derive New Key	Reset, self_test serial command, Periodic
Round Key Expansion	AES key expands into round keys	KAT	Critical Function	Serial Response Message, HW Error LED=Off, HW Error pin='0'	Key Expansion Test	Reset, self_test serial command, Periodic
DRBG	Instantiate and Generate	KAT	CAST	Serial Response Message, HW Error LED=Off, HW Error pin='0'	DRBG Test	Reset, self_test serial command, Periodic
Entropy Continuous Health Tests	False positive rates of 2^-20 (APT) and 2^-50 (RCT)	APT and RCT	Critical Function	Serial Response Message, HW Error LED=Off, HW Error pin='0'	Health Tests	Reset, self_test serial command, Periodic
AES-ECB Encrypt	128	KAT	CAST	Serial Response Message, HW Error	Encryption known answer test	Reset, self_test serial

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
				LED=Off, HW Error pin='0'		command, Periodic

Table 21: Conditional Self-Tests

10.3 Periodic Self-Test Information

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
CRC32	EDC	SW/FW Integrity	Every time the module is reset	Reset the module
Entropy Start-up Health Tests	APT and RCT	Critical Function	Every time the module is reset	Reset the module

Table 22: Pre-Operational Periodic Information

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
SHA3	KAT	CAST	24 hours	Programmatically, self_test Serial Command Message
AES Encrypt 1	Comparison	CAST	Continuous	
AES Encrypt 2	Comparison	CAST	Continuous	
AES Decrypt 1	Comparison	CAST	Continuous	
AES Decrypt 2	Comparison	CAST	Continuous	
KBKDF KMAC	KAT	CAST	24 hours	Programmatically, self_test Serial Command Message
Round Key Expansion	KAT	Critical Function	24 hours	Programmatically, self_test Serial Command Message
DRBG	KAT	CAST	24 hours	Programmatically, self_test Serial Command Message
Entropy Continuous Health Tests	APT and RCT	Critical Function	Continuous	Programmatically, self_test or get_trn Serial Command Message.
AES-ECB Encrypt	KAT	CAST	24 hours	Programmatically, self_test Serial Command Message

Table 23: Conditional Periodic Information

The Datapath Encryption and Datapath Decryption services are implemented as two parallel instances as shown in Figure 3. The output ciphertext blocks of the two parallel Encryption implementations are compared on a block-by-block basis. If the comparison fails the module enters the HW Error State. Similarly, the output plaintext blocks of the two parallel Decryption implementations are compared on a block-by-block basis. If the comparison fails, the module enters the HW Error state. This continual test meets the periodic test requirement per 140-3 IG10.3.E

Periodic tests are run every 24 hours when the module is the Approved state. For the Entropy/DBRG this is performed using the ATECC608C Self-Test command. Also, the Entropy/DRBG test is performed any time the operator requests a random number using the get_trn serial command message.

10.4 Error States

Name	Description	Conditions	Recovery Method	Indicator
Init Error	Module cannot initialize	FPGA configuration error	Reset the module	After Reset, Init Error pin remains asserted and Init Error LED remains on
HW Error	Any self-test failure	Powerup self-test failure, Conditional self-test failure, Periodic self-test failure	Reset the module	HW Error Pin Asserted, HW Error LED Illuminated, Fault Status Serial Message
EFP Error	EFP	Supply voltages or FPGA die temperature outside of approved range	Reset the module	HW Error Pin Asserted, HW Error LED Illuminated, Fault Status Serial Message

Table 24: Error States

10.5 Operator Initiation of Self-Tests

There are two methods for the operator to initiate Self-Test:

- Reset the module. This will complete the pre-operational self-tests and conditional self-tests before the module enters normal operation in the Approved state.
- Send the self_test serial command message to the Control Input serial message pin. This will stop all cryptographic services, perform all conditional self-tests, then resume all cryptographic services.

10.6 Additional Information

If any self-test fails, the module will enter the HW Error State. Before entering the HW Error State, the module will store an error log and then send out a get_status serial response message out the CpuRd pin. The format of the get_status response message is shown in the table below.

The stored error log (bits 32-8 in the get_status response message) is persistent through reset and power cycles. This allows the operator to determine what error condition or conditions occurred. If the module can be recovered from the HW Error State, then the operator can use the get_status serial command message and examine the logged errors in the response. The operator can then clear the error log by using the self_test serial command message, which will perform the self-test and clear the error log.

Status Bit	Name	Description
7-0	Response Code	0x21 = get_status response message.
23-8	Stored Error Log	Stored Error Log (Stored when entering the Error State) Bit 23: CO Authentication Not Loaded Bit 22: Operational / Operational LED On Bit 21: HW Fault Error / Fault LED On Bit 20: 3.3V Power Supply Out of Range Bit 19: 2.5V Power Supply Out of Range Bit 18: 1.8V Power Supply Out of Range Bit 17: 1.0V Power Supply Out of Range Bit 16: Temperature Out of Range Bit 15: ATECC608C Cryptographic IC Fault Bit 14: SHA3-256 Self-Test Error Bit 13: KMAC256 Self-Test Error Bit 12: KDF KMAC Self-Test Error Bit 11: AES-CBC3 Encryption Self-Test Error Bit 10: AES-CBC3 Encryption Comparison Test Error Bit 9: AES-CBC3 Decryption Self-Test Error Bit 8: AES-CBC3 Decryption Comparison Test Error
39-24	Current Errors	Bit 39: CO Authentication Not Loaded Bit 38: Operational / Operational LED On Bit 37: HW Fault Error / Fault LED On Bit 36: 3.3V Power Supply Out of Range Bit 35: 2.5V Power Supply Out of Range Bit 34: 1.8V Power Supply Out of Range Bit 33: 1.0V Power Supply Out of Range Bit 32: Temperature Out of Range Bit 31: ATECC608C Cryptographic IC Fault Bit 30: SHA3-256 Self-Test Error Bit 29: KMAC256 Self-Test Error Bit 28: KDF KMAC Self-Test Error Bit 27: AES-CBC3 Encryption Self-Test Error Bit 26: AES-CBC3 Encryption Comparison Test Error Bit 25: AES-CBC3 Decryption Self-Test Error Bit 24: AES-CBC3 Decryption Comparison Test Error
53-40	VMon3v3	3.3V Power Supply Monitor Value.
67-54	VMon2v5	2.5V Power Supply Monitor Value.
81-68	Vaux	1.8V Power Supply Monitor Value.
95-82	VInt	1.0V Power Supply Monitor Value.
109-96	Temp	FPGA Junction temperature monitor value.
110	SelfTestRd	ATECC608C Self-Test Complete Since Last get_status Command.
111	TestActive	Module in Factory Test Mode.

Status Bit	Name	Description
112	FlashLocked	Bitstream File / Flash Write Protected.
113	CfgLocked	ATECC608C Configuration Locked / Write Protected.
114	OtpLocked	ATECC608C OTP Memory Locked.
118-115	ERollRqstBus	Encryption Channel 4:1 KeyRoll requested.
122-119	E-RK/IV Alarms	Encryption Channel 4:1 round key or IV buffer empty
126-123	EKeyZeroBus	Encryption Channel 4:1 KDK is Zero.
157-126	D-RK/IV Alarms	Decryption Channel 32:1 round key or IV buffer empty
189-158	EKeyZeroBus	Encryption Channel 4:1 KDK is Zero.

Table 25: Status Response Message Bit Mapping

11 Life-Cycle Assurance

11.1 Installation, Initialization, and Startup Procedures

The Stinger Cryptographic Module is factory installed into Datum radio products. End-user installation, maintenance and replacement are not supported.

Initialization is performed by the Datum radio motherboard on powerup by resetting the module. After the power-up self-test is complete, as indicated by the HW Error pin on the Status Output interface, the module should have its hardware and firmware versions verified by get_id serial command message. The module should match the information provided in Table 3. If the information does not match, the Datum radio and module should immediately be taken out of service and returned to Datum Systems.

Before first use or after module zeroization, the module will require that an operator setup the User and CO passwords using the set_user and set_co serial command messages to access approved services requiring authentication. A module with existing User and CO credentials can be zeroized to erase all SSPs and allow a new operator to establish their own authentication credentials. If a module is taken out of service and/or repurposed, it can be zeroized, erasing all SSPs. The module requires no maintenance or periodic inspections.

11.2 Administrator Guidance

There is no administrator guidance needed beyond the services available through the module's serial message protocol.

11.3 Non-Administrator Guidance

There is no non-administrator guidance needed beyond the services available through the module's serial message protocol.

11.4 End of Life

To securely sanitize the module, the operator should perform the zero_all serial command message. This will zeroize all SSPs in the module and return it to the factory state.

After end of life, users can return the entire Datum radio to Datum Systems for recycling. If users want to securely destroy a module, The Datum Radio should be disassembled and the module printed circuit board assembly should be shredded using an electronics shredder designed to destroy storage media such as solid-state and hard disk drives.

12 Mitigation of Other Attacks

The module continuously monitors the supply voltages and junction temperature within the Artix7 FPGA. If any of the supply voltages an/or the junction temperature fall outside the acceptable range, the module enters the EFP Error state. In the EFP Error state all cryptographic services, the Data Input, Data Output, and Control Input interfaces are inoperable. The only way to exit the EFP Error state is to reset the module, with or without a power-cycle.

When in the EFP Error state, no cryptographic services are available and the module periodically sends either on OK or ERR response message. The ERR response message is sent as long as one or more of the voltages or junction temperature measurements fall outside the acceptable range. When all of the voltages and the junction temperature return to the normal range, the OK response message is sent. The operator can monitor the response messages to determine when it may be possible to reset and restart the module. The only way to exit the EFP Error state is to reset the module, with or without power-cycle.

Additionally, if the FPGA junction temperature exceeds +125C, the FPGA will automatically de-configure to prevent damage to the device. When de-configured, the FPGA has no logic active and is in the Init Error state. No cryptographic services are available. The only way to exit the Init Error state is to reset the module, with or without power-cycle.

The normal operational voltage and temperature ranges are shown in the table below.

Temp/Voltage Type	Temperature or Voltage measurement	Specify EFP or EFT	Specify if this condition results in a shutdown or zeroization
Junction Low Temperature	<= -40C	EFP	Shutdown (EFP Error State)
Junction High Temperature	>= +100C	EFP	Shutdown (EFP Error State)
Low Voltage (1.0V supply)	<= 0.950V	EFP	Shutdown (EFP Error State)
High Voltage (1.0V supply)	>= 1.050V	EFP	Shutdown (EFP Error State)
Low Voltage (1.8V supply)	<= 1.710V	EFP	Shutdown (EFP Error State)
High Voltage (1.8V supply)	>= 1.890V	EFP	Shutdown (EFP Error State)
Low Voltage (2.5V supply)	<= 2.375V	EFP	Shutdown (EFP Error State)
High Voltage (2.5V supply)	>= 2.625V	EFP	Shutdown (EFP Error State)
Low Voltage (3.3V supply)	<= 3.135V	EFP	Shutdown (EFP Error State)
High Voltage (3.3V supply)	>= 3.465V	EFP	Shutdown (EFP Error State)

Table 26: EFP/EFT Information

13 References and Definitions

List with the different references and definitions used in this document.

Abbreviation	Full Specification Name
[NIST]	National Institute of Standards and Technology
[FIPS140-3]	Security Requirements for Cryptographic Modules, March 22, 2019
[IG]	Implementation Guidance for FIPS PUB 140-3 and the Cryptographic Module Validation Program
[ISO19790]	Information technology – Security techniques – Security requirements for cryptographic modules, 2012(2014)
[38A]	NIST Special Publication 800-38A, Recommendation for Block Cipher Modes of Operation, December 2001
[38B]	NIST Special Publication 800-38B, Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication, May 2005