Geotab Inc.

Geotab Extended Cryptographic Module

FIPS 140-3 Non-Proprietary Security Policy

GEOTAB
management by measurement

Table of Contents

List of Tables

List of Figures

1 General

1.1 Overview

This document is the non-proprietary security policy for the Geotab Extended Cryptographic Module from Geotab Inc. This Security Policy describes how the Geotab Extended Cryptographic Module meets the security requirements of Federal Information Processing Standards (FIPS) Publication 140-3 for security level 1.

1.2 Security Levels

The following table lists the level of validation for each area in FIPS-140-3

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

1.3 Additional Information

Reference	Description
[140-3 IG]	FIPS 140-3 Implementation Guidance
[RFC 8446]	RFC 8446

2 Cryptographic Module Specification

2.1 Description

Module description
The Geotab Cryptographic module is classified as **Level 1 Firmware Module** as per FIPS-140-3 guidelines executing within a limited operational environment. The module operates within a

single-chip module embodiment. The Tested Operational Environment's physical perimeter is a single-chip microcontroller. The module is bundled with the firmware image used by the physical microcontrollers. The module performs no communication other than with the calling application via well-defined APIs that invoke the module.

System description

The system described by this security policy is a single-chip firmware module, running in a limited operational environment. The operating environment has an operating system and contains a single CPU with one core. The environment is multithreaded. Program instructions of the module and volatile SSPs are stored within the cryptographic boundary. The module does not provide non-volatile key storage. Keys are provided externally by the application layer and are not stored by the module in non-volatile memory. Raw entropy is provided externally from an accelerometer and is conditioned within the module, or uses a pre-conditioned entropy source. The Conditioners used being SHA2-256/AES-CMAC.
Module Type: Firmware

Module Embodiment: Single Chip

Module Characteristics:

Module Boundary

The following block diagram details modules' Tested Operational Environment's Physical Perimeter and Cryptographic boundaries.

Cryptographic Boundary

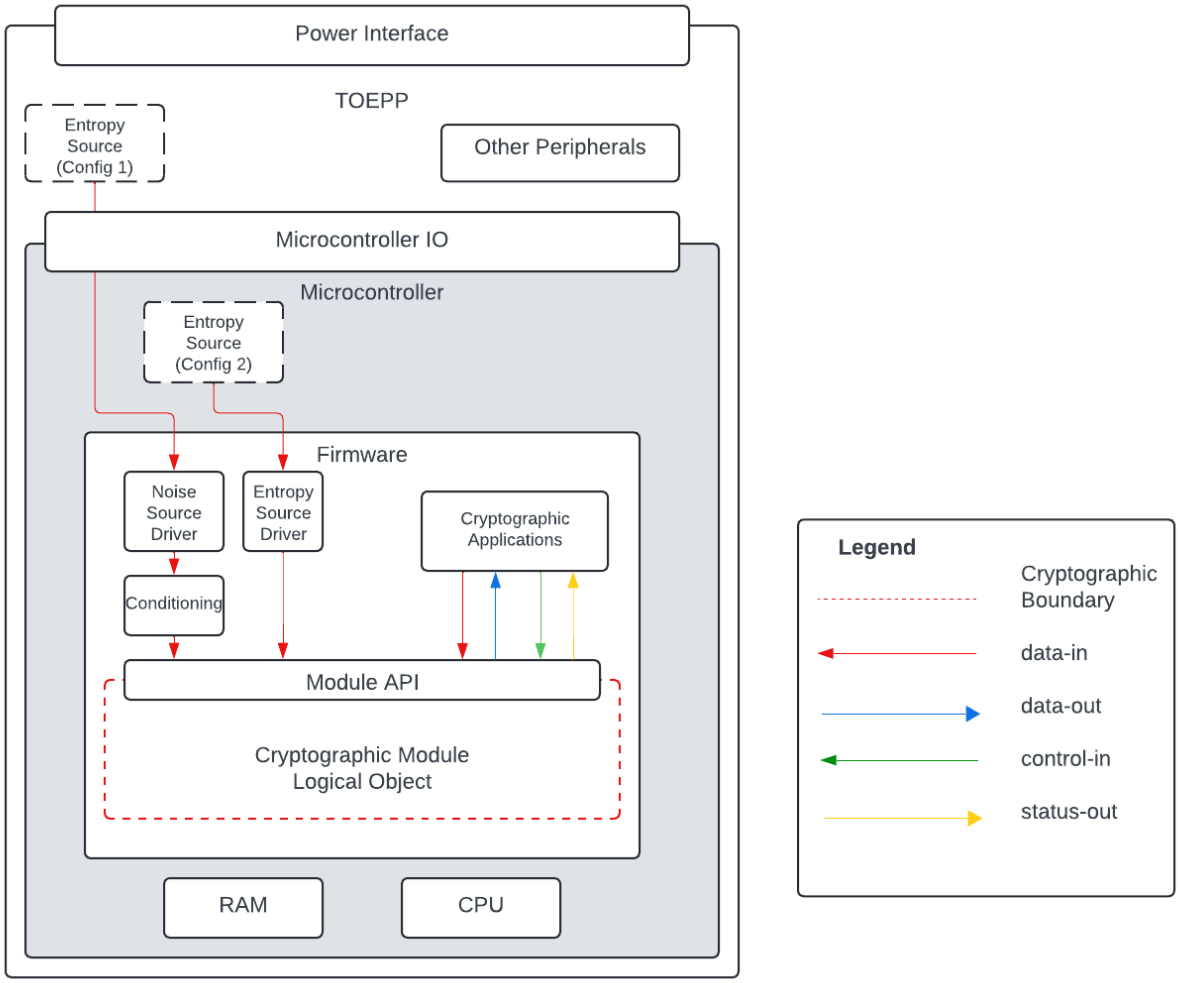

Figure 1. Cryptographic Boundary

Tested Operational Environment's Physical Perimeter (TOEPP):

Per figure 1 for Configuration 1, the Tested Operational Environment's physical perimeter is the S32K Core Telematics Board and STM32H7 Core Telematics Board that the module operates on.

Per figure 1 for Configuration 2, the Tested Operational Environment's physical perimeter is the STM32U5 Core Telematics Board that the module operates on.

2.2 Tested and Vendor Affirmed Module Version and Identification

The Tested Module is a Geotab Cryptographic module (Firmware Module Level 1) and its identification is stated below.

Tested Module Identification – Hardware:

N/A for this module.

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

Package or File Name	Software/ Firmware Version	Features	Integrity Test
fipslib.bin	1.0	Compiled for ARM Cortex M33 on STM32U5 (ARMv8-M) - No PAA	Yes
fipslib.bin	1.0	Compiled for ARM Cortex M33 on STM32U5 (ARMv8-M) - PAA	Yes
fipslib.bin	1.0	Compiled for ARM Cortex M4F on NXP S32K14X (Armv7E-M)	Yes
fipslib.bin	1.0	Compiled for ARM Cortex M7 on STM32H7 (Armv7E-M)	Yes

Table 2: Tested Module Identification – Software, Firmware, Hybrid (Executable Code Sets)

The product has been tested and is intended to operate on the following Geotab Operating Environments. There are no Vendor affirmed operational environments.

Tested Module Identification – Hybrid Disjoint Hardware:

N/A for this module.

Tested Operational Environments - Software, Firmware, Hybrid:

Operating System	Hardware Platform	Processors	PAA/PAI	Hypervisor or Host OS	Version(s)
Apache Nuttx	S32K Core Telematics Board	NXP S32K14X (Armv7E-M)	No		1.0
Apache Nuttx	STM32H7 Core Telematics Board	STM32H7 (Armv7E-M)	No		1.0
Apache Nuttx	STM32U5 Core Telematics Board	STM32U5 (Armv8-M)	No		1.0
Apache Nuttx	STM32U5 Core Telematics Board	STM32U5 (Armv8-M)	Yes		1.0

Table 3: Tested Operational Environments - Software, Firmware, Hybrid

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

N/A for this module.

2.3 Excluded Components

There are no excluded components

2.4 Modes of Operation

Modes List and Description:

The following are the modes of operation and their description mentioned in the Table below.

Mode Name	Description	Type	Status Indicator
Approved Mode of Operation	The mode of operation in which the module is operating correctly	Approved	Successful completion of service

Table 4: Modes List and Description

The module does not support any operation in a degraded mode.
In the approved mode of operation, the *"Get Version"* service will return an integer value corresponding to the module revision which is 1.0.

2.5 Algorithms

Approved Algorithms:

Algorithm	CAVP Cert	Properties	Reference
AES-CBC	A5265	Direction - Decrypt, Encrypt Key Length - 256	SP 800-38A
AES-GCM	A5265	Direction - Decrypt, Encrypt IV Generation - External Key Length - 128, 256 Tag Length - 128 IV Length - IV Length: 96 Payload Length - Payload Length: 0-65536 Increment 8 AAD Length - AAD Length: 0-65536 Increment 8	SP 800-38D
ECDSA KeyGen (FIPS186-5)	A5265	Curve - P-224, P-256, P-384 Secret Generation Mode - extra bits	FIPS 186-5
ECDSA KeyVer (FIPS186-5)	A5265	Curve - P-224, P-256, P-384	FIPS 186-5
ECDSA SigGen (FIPS186-5)	A5265	Curve - P-224, P-256, P-384 Hash Algorithm - SHA2-256, SHA2-384 Component - No	FIPS 186-5
ECDSA SigVer (FIPS186-5)	A5265	Curve - P-224, P-256, P-384 Hash Algorithm - SHA2-256, SHA2-384	FIPS 186-5
HMAC DRBG	A5265	Prediction Resistance - No Mode - SHA2-256	SP 800-90A Rev. 1
HMAC-SHA2-256	A5265	Key Length - Key Length: 112-512 Increment 8	FIPS 198-1
RSA SigVer (FIPS186-5)	A5265	Modulo - 2048, 3072 Signature Type - pkcs1v1.5	FIPS 186-5
SHA2-256	A5265	Message Length - Message Length: 0-51200 Increment 8	FIPS 180-4
SHA2-384	A5265	Message Length - Message Length: 0-51200 Increment 8	FIPS 180-4
TLS v1.3 KDF (CVL)	A5265	HMAC Algorithm - SHA2-256 KDF Running Modes - PSK-DHE	SP 800-135 Rev. 1

Table 5: Approved Algorithms

Note: Only the algorithms specified in the table above are supported by the module in approved mode of operation

Vendor-Affirmed Algorithms:

Name	Properties	Implementation	Reference
CKG	Key Type:Asymmetric	N/A	Section 4 Example 1

Table 6: Vendor-Affirmed Algorithms

Non-Approved security functions

The module does not support any of the following:
“Non-Approved, Allowed Algorithms”,
"Non-Approved Algorithms Allowed in the Approved Mode of Operation with No Security Claimed".
“Non-Approved, Not Allowed Algorithms”

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
AES-CBC	BC-UnAuth	Used for Authentication & encryption		AES-CBC: (A5265)
AES-GCM	BC-Auth	Used for Authentication & encryption		AES-GCM: (A5265)
DRBG	DRBG	Deriving Entropy		HMAC DRBG: (A5265)
HMAC	MAC	Integrity Check		HMAC-SHA2-256: (A5265)
KeyGen	AsymKeyPair-KeyGen	Key Pair Generation		ECDSA KeyGen (FIPS186-5): (A5265)
KeyVer	AsymKeyPair-PubKeyVal	Public Key Verification		ECDSA KeyVer (FIPS186-5): (A5265)
SHA	SHA	Hash Function		SHA2-256: (A5265) SHA2-384: (A5265)

Name	Type	Description	Properties	Algorithms
SigGen	DigSig-SigGen	Signature Generation		ECDSA SigGen (FIPS186-5): (A5265)
SigVer	DigSig-SigVer	Signature Verification		ECDSA SigVer (FIPS186-5): (A5265) RSA SigVer (FIPS186-5): (A5265)
TLSv1.3	KAS-135KDF	Key Agreement		TLS v1.3 KDF: (A5265)

Table 7: Security Function Implementations

The Security Function implementations are mentioned in the table above.

2.7 Algorithm Specific Information

IG C.H Conformance

1) The module itself does not implement the TLS protocol. The module's implementation of AES-GCM is used by an application that runs outside the module's cryptographic boundary. Per the design of the TLS protocol, the counter (the nonce_explicit part of the IV) does not exhaust the maximum number of possible values for a given session key. In case the counter exhausts, the module returns an error indication to the calling application, which will then need to either abort the connection, or trigger a handshake to establish a new encryption key.
   This implementation falls under Scenario 1 of FIPS 140-3 IG C.H. The protocol that provides this compliance is TLS 1.3, defined in RFC8446 of August 2018, using the cipher-suites that explicitly select AES GCM as the encryption/decryption cipher (Appendix B.4 of RFC8446). The module supports acceptable AES GCM cipher suites from Section 3.3.1 of SP800-52r2. TLS 1.3 employs separate 64-bit sequence numbers, one for protocol records that are received, and one for protocol records that are sent to a peer. These sequence numbers are set at zero at the beginning of a TLS 1.3 connection and each time when the AES-GCM key is changed. After reading or writing a record, the respective sequence number is incremented by one. The protocol specification determines that the sequence number should not wrap, and if this condition is observed, then the protocol implementation must either trigger a re-key of the session (i.e., a new key for AES-GCM), or terminate the connection. Utilizing AES-GCM outside of this intended use (TLS v1.3) is use of the service and Module outside of an approved manner.

2.8 RBG and Entropy

RBG entropy sources

The entropy source resides outside the module's cryptographic boundary but within the module's physical perimeter.

Cert Number	Vendor Name
E86	Geotab Inc.
E11	STMicroelectronics

Table 8: Entropy Certificates

Name	Type	Operational Environment	Sample Size	Entropy per Sample	Conditioning Component
Accelerometer noise source	Physical	Apache Nuttx on an NXP S32K14X running on an Armv7E-M and Apache Nuttx on an STMicroelectronics STM32H7 running on an Armv7E-M	256 bits	256 bits	SHA256 conditioning component (A5265)
STM32U5x TRNG	Physical	STM32U5x advanced Arm®-based 32-bit MCU	128 bits	128 bits	AES-CMAC conditioning component (A1729)

Table 9: Entropy Sources

2.9 Key Generation

The module generates asymmetric keys per section 5.1 of SP 800-133r2 as part of the 'Asymmetric Generate Keypair' service.
The asymmetric ECDSA keys are generated per the requirements of FIPS 186-5.

2.10 Key Establishment

This module does not support any Key Agreement Schemes.

2.11 Industry Protocols

None

3 Cryptographic Module Interfaces

3.1 Ports and Interfaces

The physical ports of the module are the same as the special purpose single chip microcontroller system on which it is executing. The logical interface is an application programming interface (API), the mapping of logical interface type is explained below.

Physical Port	Logical Interface(s)	Data That Passes
N/A	Data Input	API entry point data input parameters
N/A	Data Output	API entry point data output parameters
N/A	Control Input	API entry point and corresponding parameters
N/A	Status Output	API entry point return values
N/A	Power	N/A

Table 10: Ports and Interfaces

Since the module boundary is firmware, control of the physical ports is outside the module scope. When the module is in self-test state, error state or zeroization state, all output on the logical data output interface is prohibited. In error state module will only return error value (no data output is returned).

4 Roles, Services, and Authentication

4.1 Authentication Methods

N/A for this module.

The module does not support operator authentication.

4.2 Roles

The module supports following role:

Name	Type	Operator Type	Authentication Methods
Crypto Officer (CO)	Role	Crypto Officer	None

Table 11: Roles

The module supports the Crypto Officer (CO) role and does not support multiple concurrent roles or any maintenance or bypass capability. The CO role is implicitly assumed by the application accessing services implemented by the module.

4.3 Approved Services

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Initialize Module	Initializes the module and runs the designated self tests.	Successful completion of service (Status = 0)	Module Integrity Value	Status	None	Crypto Officer (CO) - Module Integrity Value: W,E
Asymmetric Generate Keypair	Generates Key pairs.	Successful completion of service (Status = 0)	Flags	Status, Generated Key Pair - Private, Generated Key Pair - Public	KeyGen	Crypto Officer (CO) - Generated Key Pair - Private: G,R - Generated Key Pair - Public: G,R
Asymmetric Keypair Verification	Verify the Keypair	Successful completion of service (Status = 0)	Verify the Keypair	Verify: Status	KeyVer	Crypto Officer (CO) - Digital Signature - Verification Key: W,E - AES-GCM IV: W
Digital Signature	Generate or verify digital signatures	Successful completion of service (Status = 0)	Sign: Digital Signature - Generation Key Verify: Digital Signature - Verification Key,	Sign: Status, signature value. Verify: status	SigGen SigVer	Crypto Officer (CO) - Digital Signature - Generation Key: W,E - Digital Signature -

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						Verification Key: W,E
Get State	Provide Module state	Module state is provided.		Module State		Crypto Officer (CO)
Get Version	Provides Module Version	Module Version is provided.		Module Version		Crypto Officer (CO)
Key Derivation	Derive key material from a shared secret to be exported from the module.	Successful completion of service (Status = 0)	Shared Secret	Status, Derived Key Material	TLSv1.3	Crypto Officer (CO) - Derived Key Material: G,R - Shared Secret: W,E
Keyed Hash	Generate or verify message integrity	Successful completion of service (Status = 0)	Message, Keyed Hash Key	Status, Hash Value	HMAC	Crypto Officer (CO) - Keyed Hash Key: W,E
Message Digest	Generate a message digest	Successful completion of service (Status = 0)	Message	Status, Hash Value	SHA	Crypto Officer (CO)
Random	Generate random bits using a DRBG	Successful completion of service (Status = 0)	Number of bits to generate, Entropy Seed	Status, randomly generated bits.	DRBG	Crypto Officer (CO) - Symmetric Cipher Key - CBC: G,R - Symmetric Cipher

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						Key - GCM: G,R
Symmetric Ciphers	Encrypt or Decrypt data with AES including CBC, and GCM modes.	Successful completion of service (Status = 0)	Symmetric Cipher Key	Status, Plaintext or Ciphertext data	AES-CBC AES-GCM	Crypto Officer (CO) - Symmetric Cipher Key - CBC: W,E - Symmetric Cipher Key - GCM: W,E
Zeroize	Unitialize any context and zeroise any SSPs. These calls are in the format "fips140_xxxFree"	Successful completion of service (Status = 0)	Context to zeroise	Status	None	Crypto Officer (CO) - Derived Key Material: Z - Digital Signature - Generation Key: Z - Digital Signature - Verification Key: Z - Entropy Input String: Z - Entropy Seed: Z - Generated Key Pair - Private: Z -

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						Generated Key Pair - Public: Z - Keyed Hash Key: Z - Module Integrity Value: Z - Secret C: Z - Secret V: Z - Shared Secret: Z - Symmetric Cipher Key - CBC: Z - Symmetric Cipher Key - GCM: Z

Table 12: Approved Services

The indicator field of the above table specifies the “successful completion of service” means Services return success status; non-zero value or the approved security function finishes successfully.

4.4 Non-Approved Services

N/A for this module.

4.5 External Software/Firmware Loaded

N/A for this module.

5 Software/Firmware Security

5.1 Integrity Techniques

The firmware module is in the form of a compiled binary called fipslib.bin which is a subset of the overall firmware binary. (geotab.bin = businesslogic.bin + fipslib.bin)

After a module is powered on, the module's integrity check using HMAC-SHA2-256 (#A5265) is performed during module instantiation. If the Integrity check fails, it will cause the module to enter an error state. All the temporary values generated during the integrity check are zeroized upon completion of the operation.

Additionally, the module performs listed self-tests and if successful the module is deemed to be usable by the user. If the self-tests or HMAC based integrity test fails for any reason the module will only provide the 'Initialize Module', 'Get State', and 'Get Version' services.

5.2 Initiate on Demand

The operator is able to initiate the integrity test on demand by re-initializing the module or by power-cycling the host platform. Upon initialization the module performs an integrity test of the module on itself using HMAC-SHA2-256 (#A5265). If the Integrity check fails, it will cause the module to enter an error state.

6 Operational Environment

6.1 Operational Environment Type and Requirements

Type of Operational Environment: Limited

How Requirements are Satisfied:

The Module tested operating environments are listed below. The module functions entirely within the Operating environment provided space for the calling application.

- S32K Core Telematics Board with the NXP S32K14X (Armv7E-M) processor
- STM32H7 Core Telematics Board with the STM32H7 (Armv7E-M) processor
- STM32U5 Core Telematics Board with the STM32U5 (Armv8-M) processor

7 Physical Security

The module implements standard passivation for its components.
The module employs production-grade components.

8 Non-Invasive Security

This cryptographic module does not claim any non-Invasive security features.

9 Sensitive Security Parameters Management

9.1 Storage Areas

Storage Area Name	Description	Persistence Type
RAM	Volatile Memory area where all SSPs are stored	Dynamic

Table 13: Storage Areas

All Sensitive Security Parameters (SSPs) are stored in volatile memory in plaintext format.

9.2 SSP Input-Output Methods

Name	From	To	Format Type	Distribution Type	Entry Type	SFI or Algorithm
API Interface - Input	Volatile Memory outside the module boundary	RAM	Plaintext	Manual	Electronic	
API Interface - Output	RAM	Volatile Memory outside the module boundary	Plaintext	Manual	Electronic	

Table 14: SSP Input-Output Methods

All SSPs Input Output methods are specified in the table above.

9.3 SSP Zeroization Methods

Zeroization Method	Description	Rationale	Operator Initiation
Power Cycle	Sets the SSPs to 0 when the module is powered off.	Power loss to clear SSP values stored in volatile memory	Unplug the device

Zeroization Method	Description	Rationale	Operator Initiation
Zeroize service	Sets SSP's used by the module to 0 on request.	Need to clear temporary SSP values stored in volatile memory during runtime.	Call the Zeroize service on SFI context that is no longer needed.

Table 15: SSP Zeroization Methods

All temporary SSP values in volatile memory used are zeroized. The zeroization consists of writing zeros to the memory location used by the SSP. The zeroization service for the SSP in volatile memory consists of powering off the module, which sets all instances of SSPs to 0 value and making it non-retrievable.

9.4 SSPs

The module receives all SSPs through API calls to the module. All sensitive parameters are provided externally to the module and handled by the application using the module.

To protect the SSPs from unauthorized access, disclosure, modification, and substitution, the microcontroller's hardware Readout Protection (RDP) is set. When RDP is enabled, the module's firmware and the confidentiality and integrity of all stored SSPs are protected.

The module supports the following SSP’s listed below

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
AES-GCM IV	Initialization Vector for AES-GCM operation	96 bits - 96 bits	Initialization Vector - PSP			AES-GCM
Derived Key Material	Key Derivation derived key material. TLS KDF v1.3:	256, 384 bits - 128, 192 bits	Derived Key Material - CSP	TLSv1.3		TLSv1.3
Digital Signature - Generation Key	Private key for signature generation. Supported Algorithm: ECDSA	P-224, P-256, P-384 bits - 112, 128, 192 bits	Asymmetric Key - CSP	KeyGen		SigGen
Digital Signature -	Public key for signature verification.	P-224, P-256, P-384,	Asymmetric Key - CSP	KeyGen		SigVer

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
Verification Key	Supported Algorithms: RSA, ECDSA	2048, and 3072 bits - 112, 128, 192 bits and 112, 128 bits				
Entropy Input String	Entropy input string from external entropy source.	1024 bits - 1024 bits	Entropy - CSP			DRBG
Entropy Seed	Seed entropy taken from Entropy Input String	128-bits - 128-bits	Entropy - CSP	DRBG		DRBG
Generated Key Pair - Private	Generated private key component Algorithm: ECDSA	P-224, P-256, P-384 bits - 112, 128, 192 bits	Asymmetric Key - CSP	KeyGen		KeyGen
Generated Key Pair - Public	Generated Public key component. Supported Algorithm: ECDSA	P-224, P-256, P-384 bits - 112, 128, 192 bits	Asymmetric Key - CSP	KeyGen		KeyGen
Keyed Hash Key	Key for keyed hash service.	112-512 bits - 112 bits (for 112 bit keys), 128 bits (for >112 bit keys)	Symmetric Key - CSP	DRBG		HMAC
Module Integrity Value	Pre-calculated value for module integrity.	256 bits - 256 bits	Message Authentication - Neither	Pre-loaded		HMAC
Secret C	Hash DRBG Internal	256 bits - 256 bits	Entropy - CSP	DRBG		DRBG

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
	state secret C					
Secret V	Hash DRBG Internal state secret V	256 bits - 256 bits	Entropy - CSP	DRBG		DRBG
Shared Secret	Shared secret for deriving Derived Key Material. Algorithm: ECDSA	P-224, P-256, P-384 bits - 112, 128, 192 bits	Shared Secret - CSP			TLSv1.3
Symmetric Cipher Key - CBC	AES Key used for encryption and decryption operations for AES CBC mode	256-bits - 256-bits	Symmetric key - CSP	DRBG		AES-CBC
Symmetric Cipher Key - GCM	AES Key used for encryption and decryption operations. For AES GCM mode	128 and 256 bits - 128 and 256 bits	Symmetric key - CSP	DRBG		AES-GCM

Table 16: SSP Table 1

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
AES-GCM IV	API Interface - Input	RAM:Plaintext	Duration of use	Power Cycle Zeroize service	
Derived Key Material	API Interface - Output	RAM:Plaintext	Duration of use	Power Cycle Zeroize service	Shared Secret:Derived From
Digital Signature -	API Interface - Input	RAM:Plaintext	Duration of use	Power Cycle	Generated Key Pair - Private:Paired With

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
Generation Key				Zeroize service	
Digital Signature - Verification Key	API Interface - Input	RAM:Plaintext	Duration of use	Power Cycle Zeroize service	Generated Key Pair - Public:Paired With
Entropy Input String	API Interface - Input	RAM:Plaintext	Duration of use	Power Cycle	
Entropy Seed	API Interface - Input	RAM:Plaintext	Duration of use	Power Cycle Zeroize service	Entropy Input String:Derived From
Generated Key Pair - Private	API Interface - Output	RAM:Plaintext	Duration of use	Power Cycle Zeroize service	Digital Signature - Generation Key:Paired With
Generated Key Pair - Public	API Interface - Output	RAM:Plaintext	Duration of use	Power Cycle Zeroize service	Digital Signature - Verification Key:Paired With
Keyed Hash Key	API Interface - Input	RAM:Plaintext	Duration of use	Power Cycle Zeroize service	
Module Integrity Value		RAM:Plaintext	Duration of use	Power Cycle Zeroize service	
Secret C		RAM:Plaintext	Duration of use	Power Cycle Zeroize service	Entropy Seed:Derived From
Secret V		RAM:Plaintext	Duration of use	Power Cycle Zeroize service	Entropy Seed:Derived From
Shared Secret	API Interface - Input	RAM:Plaintext	Duration of use	Power Cycle Zeroize service	
Symmetric Cipher Key - CBC	API Interface - Input	RAM:Plaintext	Duration of use	Power Cycle Zeroize service	

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
Symmetric Cipher Key - GCM	API Interface - Input	RAM:Plaintext	Duration of use	Power Cycle Zeroize service	

Table 17: SSP Table 2

10 Self-Tests

If any self-test fails, the module enters an error state. To attempt to clear the error, re-initialize the module. A power cycle is required in order to run power on self-tests on demand.

10.1 Pre-Operational Self-Tests

The Pre-operational self-test is listed in the table below. The algorithm used in the integrity test i.e. HMAC SHA2-256 is tested before it is used in the integrity test.

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details
HMAC-SHA2-256 (A5265)	256-bits key length	KAT	SW/FW Integrity	Return success	Integrity test performed on the module image. Integrity verified using HMAC SHA-256.

Table 18: Pre-Operational Self-Tests

The module does not support bypass.
The module does not implement any pre-operational critical function tests.

10.2 Conditional Self-Tests

All the Conditional Cryptographic Algorithm Tests (CAST) are listed here in the table below. All CASTs must pass before the module can operate.

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
Adaptive Proportion Test	N/A	Fault Detection	CAST	Returns success	Checks the proportion of ones and zeros in consecutive DRBG output blocks.	Entropy Collection

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
AES-CBC Decrypt (A5265)	Key length - 256 bits	KAT	CAST	Returns success	Decryption	Power-on
AES-CBC Encrypt (A5265)	Key length - 256 bits	KAT	CAST	Returns success	Encryption	Power-on
AES-GCM Decrypt (A5265)	Key length - 256 bits	KAT	CAST	Returns success	Decrypt & ciphertext Authentication	Power-on
AES-GCM Encrypt (A5265)	Key length - 256 bits	KAT	CAST	Returns success	Encryption and provide TAG value	Power-on
ECDSA KeyGen (FIPS186-5) (A5265)	Key length - 224 bits (Based on P-224 curve)	PCT	CAST	Returns success	Key Generation	Power-on, Service Invocation
ECDSA KeyVer (FIPS186-5) (A5265)	Key length - 192 bits (based on P192 curve)	KAT	CAST	Returns success	Key Verification	Power-on
ECDSA SigGen (FIPS186-5) (A5265)	Key length - 384 bits (based on P-384 curve)	KAT	CAST	Returns success	Signature Generation	Power-on
ECDSA SigVer (FIPS186-5) (A5265)	Key length - 384 bits (based on P-384 curve)	KAT	CAST	Returns success	Signature Verification	Power-on
HMAC DRBG (A5265)	N/A	KAT	CAST	Returns success	Random Number Generator	Context Initialization
HMAC DRBG Continuous Test	N/A	Continuous test	CAST	Returns success	Confirms that the DRBG output is not stuck when new seed is requested	Service Invocation
HMAC DRBG Test Generate	N/A	KAT	CAST	Returns success	Confirms DRBG generation	Power-on
HMAC DRBG Test Instantiate	N/A	KAT	CAST	Returns success	confirm the DRBG instantiation	Power-on

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
HMAC DRBG Test Ressed	N/A	KAT	CAST	Returns success	Confirms DRBG reseed	Power-on
HMAC DRBG Test Un-instantiate	N/A	KAT	CAST	Returns success	error handling and the internal state has been zeroized	Power-on
HMAC-SHA2-256 (A5265)	256 bits	KAT	CAST	Returns success	HMAC	Power-on
Repetition Count Test	N/A	Fault Detection	CAST	Returns success	Checks for repeated output blocks from the DRBG.	Entropy Collection
RSA SigVer (FIPS186-5) (A5265)	Key length-2048 bits	KAT	CAST	Returns success	Signature Verification	Power-on
SP 800-90B Health Tests	N/A	Continuous test	CAST	Returns success	Confirms working of conditioned entropy source	Power-on
TLS v1.3 KDF (A5265)	336 bit input values, 256 bit secrets	KAT	CAST	Returns success	Key Derivation	Power-on

Table 19: Conditional Self-Tests

10.3 Periodic Self-Test Information

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
HMAC-SHA2-256 (A5265)	KAT	SW/FW Integrity	On Demand	Power cycle

Table 20: Pre-Operational Periodic Information

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
Adaptive Proportion Test	Fault Detection	CAST	On Demand	Power cycle
AES-CBC Decrypt (A5265)	KAT	CAST	On Demand	Power cycle

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
AES-CBC Encrypt (A5265)	KAT	CAST	On Demand	Power cycle
AES-GCM Decrypt (A5265)	KAT	CAST	On Demand	Power cycle
AES-GCM Encrypt (A5265)	KAT	CAST	On Demand	Power cycle
ECDSA KeyGen (FIPS186-5) (A5265)	PCT	CAST	On Demand	Power cycle
ECDSA KeyVer (FIPS186-5) (A5265)	KAT	CAST	On Demand	Power cycle
ECDSA SigGen (FIPS186-5) (A5265)	KAT	CAST	On Demand	Power cycle
ECDSA SigVer (FIPS186-5) (A5265)	KAT	CAST	On Demand	Power cycle
HMAC DRBG (A5265)	KAT	CAST	On Demand	Power cycle
HMAC DRBG Continuous Test	Continuous test	CAST	On Demand	Power cycle
HMAC DRBG Test Generate	KAT	CAST	On Demand	Power cycle
HMAC DRBG Test Instantiate	KAT	CAST	On Demand	Power cycle
HMAC DRBG Test Ressed	KAT	CAST	On Demand	Power cycle
HMAC DRBG Test Un-instantiate	KAT	CAST	On Demand	Power cycle
HMAC-SHA2-256 (A5265)	KAT	CAST	On Demand	Power cycle
Repetition Count Test	Fault Detection	CAST	On Demand	Power cycle
RSA SigVer (FIPS186-5) (A5265)	KAT	CAST	On Demand	Power cycle
SP 800-90B Health Tests	Continuous test	CAST	On Demand	Power cycle
TLS v1.3 KDF (A5265)	KAT	CAST	On Demand	Power cycle

Table 21: Conditional Periodic Information

The module does not support periodic self-testing.

10.4 Error States

Name	Description	Conditions	Recovery Method	Indicator
Module in Error State	If Integrity test or KAT fails, enter error state	N/A	Retry	Initialize Module service returns a non-zero value.

Table 22: Error States

Integrity tests and self-tests can fail and result in error status as mentioned in the table above.

11 Life-Cycle Assurance

11.1 Installation, Initialization, and Startup Procedures

Installation
The Module is already installed as part of the main firmware for the device. The device is already installed with the main firmware which includes the cryptographic module.

Initialization
The Module is initialized by calling the Initialize Module service. The initialization procedure shall be invoked by the CO Role. The CO role has responsibility to ensure that the 'Initialize Module' service returns a successful indicator before the usage of the module begins.

Startup Procedure
The device is powered on. After entering into the initialization state an integrity check is performed. Then the Module performs self-tests. Upon successful completion of the self-tests, the module enters in Operational state.

11.2 Administrator Guidance

Please contact *embeddedsecurity@geotab.com* for Administrator guidance.

11.3 Non-Administrator Guidance

The module supports the CO role and does not support non-administrators or non-administrative roles.

11.4 End of Life

The module's life is defined by the application invoking the APIs associated with the module. The module end of life is reached once the application relieves the module. The module does not store any SSPs persistently, but does store SSPs in reconfigurable memory. The module

does not store any information on non-reconfigurable memory. The procedure for secure sanitization of the module is to power it off, which is the action of zeroization of the SSPs. The sanitization via power-off, results in removal of SSP from the module.

11.5 Additional Information

Configuration Management
The code repositories are managed through Git-based configuration management and version control systems. Production grade C/C++ (gcc) compilers and assemblers are used for the development.

Platform	Hardware	Compiler	Configuration
NuttX	NXP S32K148	gcc-arm-none-eabi	-fno-builtin -Wall -Werror -Wstrict-prototypes -Wshadow -Wundef -g -mcpu=cortex-m4 -mthumb -mfloat-abi=hard -mfpu=fpv4-sp-d16 -O1 -Wl,--fatal-warnings
NuttX	ST STM32H7	arm-none-eabi-gcc	-fno-builtin -Wall -Werror -Wstrict-prototypes -Wshadow -Wundef -g -mcpu=cortex-m7 -mthumb -mfloat-abi=hard -O1 -Wl,--fatal-warnings
NuttX	ST STM32U5	arm-none-eabi-gcc	-fno-builtin -Wall -Werror -Wstrict-prototypes -Wshadow -Wundef -g -mcpu=cortex-m33 -mthumb -mfloat-abi=hard

Table 23: Compilers

Platform	Hardware	Linker	Configuration
NuttX	NXP S32K148	gcc-arm-none-eabi	n/a

NuttX	ST STM32H7	arm-none-eabi-gcc	n/a
NuttX	ST STM32U5	arm-none-eabi-gcc	n/a

Table 24: Linkers

The module does not claim any authentication functionality.

12 Mitigation of Other Attacks

The module does not claim to mitigate any additional attacks.