Envieta Systems LLC
Envieta QFlex Hardware Security Module

FIPS 140-2 Non-Proprietary Security Policy

Hardware version: 385HSM-FIPS Rev A
Hardware version: 385HSM-FIPS Rev B

Firmware version: 1.3.0
Date: June 9, 2020

Prepared by:
Acumen Security
2400 Research Blvd, Suite 395
Rockville, MD 20850
www.acumensecurity.net

Introduction

Federal Information Processing Standards Publication 140-2 — Security Requirements for Cryptographic Modules (FIPS 140-2) specifies requirements for cryptographic modules to be deployed in a Sensitive but Unclassified environment. The National Institute of Standards and Technology (NIST) and Canadian Centre for Cyber Security (CCCS) Cryptographic Module Validation Program (CMVP) run the FIPS 140 program. The NVLAP accredits independent testing labs to perform FIPS 140 testing; the CMVP validates modules meeting FIPS 140 validation. Validated is the term given to a module that is documented and tested against the FIPS 140 criteria.

More information is available on the CMVP website at: https://csrc.nist.gov/projects/cryptographic-module-validation-program

About this Document

This non-proprietary Cryptographic Module Security Policy for the QFlex Hardware Security Module (HSM) from Envieta Systems LLC provides an overview of the product and a high-level description of how it meets the overall Level 3 security requirements of FIPS 140-2.

The QFlex HSM may also be referred to as “the HSM”, or simply “the module” in this document.

Disclaimer

The contents of this document are subject to revision without notice due to continued progress in methodology, design, and manufacturing. Envieta Systems LLC shall have no liability for any error or damages of any kind resulting from the use of this document.

Notices

Table of Contents

List of Tables

List of Figures

1. Introduction

1.1 Scope

This document describes the cryptographic module security policy for the Envieta Systems LLC, Envieta QFlex Hardware Security Module (HSM) (also referred to as the “module”, “QFlex HSM" or “HSM” hereafter). It contains specification of the security rules, under which the cryptographic module operates, including the security rules derived from the requirements of the FIPS 140-2 standard. This document covers both Rev A and Rev B versions of hardware.

1.2 Overview

The QFlex HSM is a full-height, full-length, multichip embedded PCIe card that plugs into a standard PCIe Generation 3 slot, with a width of at least 8 lanes. The critical hardware components of the HSM are the Arria 10 System on Chip (SoC), which contains a dual-core ARM Cortex-A9 hard-core processor embedded in an FPGA, an Exar XR9240 cryptographic accelerator, a real time clock (RTC), and a Complex Programmable Logic Device (CPLD) based system manager. The HSM has two USB ports; one port acts as a USB master and is used to connect to a USB Smart Card reader and the second acts as a USB slave that connects to a serial UART device. Status messages will be transmitted out from this device, however no input into this device will be accepted. The use of the HSM requires a host computer running RedHat Enterprise Linux (RHEL) 7, CentOS 7, or Ubuntu 18.04 with the supporting drivers and libraries installed.

Only FIPS-Approved algorithms are implemented in the QFlex HSM. As a consequence, the QFlex HSM only supports the Approved mode of operation.

2. Security Level

The following table lists the level of validation for each area in FIPS 140-2:

FIPS 140-2 Section Title	Validation Level
Cryptographic Module Specification	3
Cryptographic Module Ports and Interfaces	3
Roles, Services, and Authentication	3
Finite State Model	3
Physical Security	3
Operational Environment	N/A
Cryptographic Key Management	3
Electromagnetic Interference / Electromagnetic Compatibility	3
Self-Tests	3
Design Assurance	3
Mitigation of Other Attacks	N/A
Overall Level	3

Table 1 – FIPS 140-2 Target Level

3. Cryptographic Module Specification

3.1 Cryptographic Boundary

The module's cryptographic boundary contains the following critical hardware components that are necessary for operation of the HSM as depicted in Figure 1 below:

- Arria 10 System-on-Chip SX-series containing an FPGA and a dual-core ARM Cortex-A9 MPCore processor
- Exar XR9240 Data Compression and Security Coprocessor
- MAX10 System Manager CPLD
- MCP79410 Real Time Clock

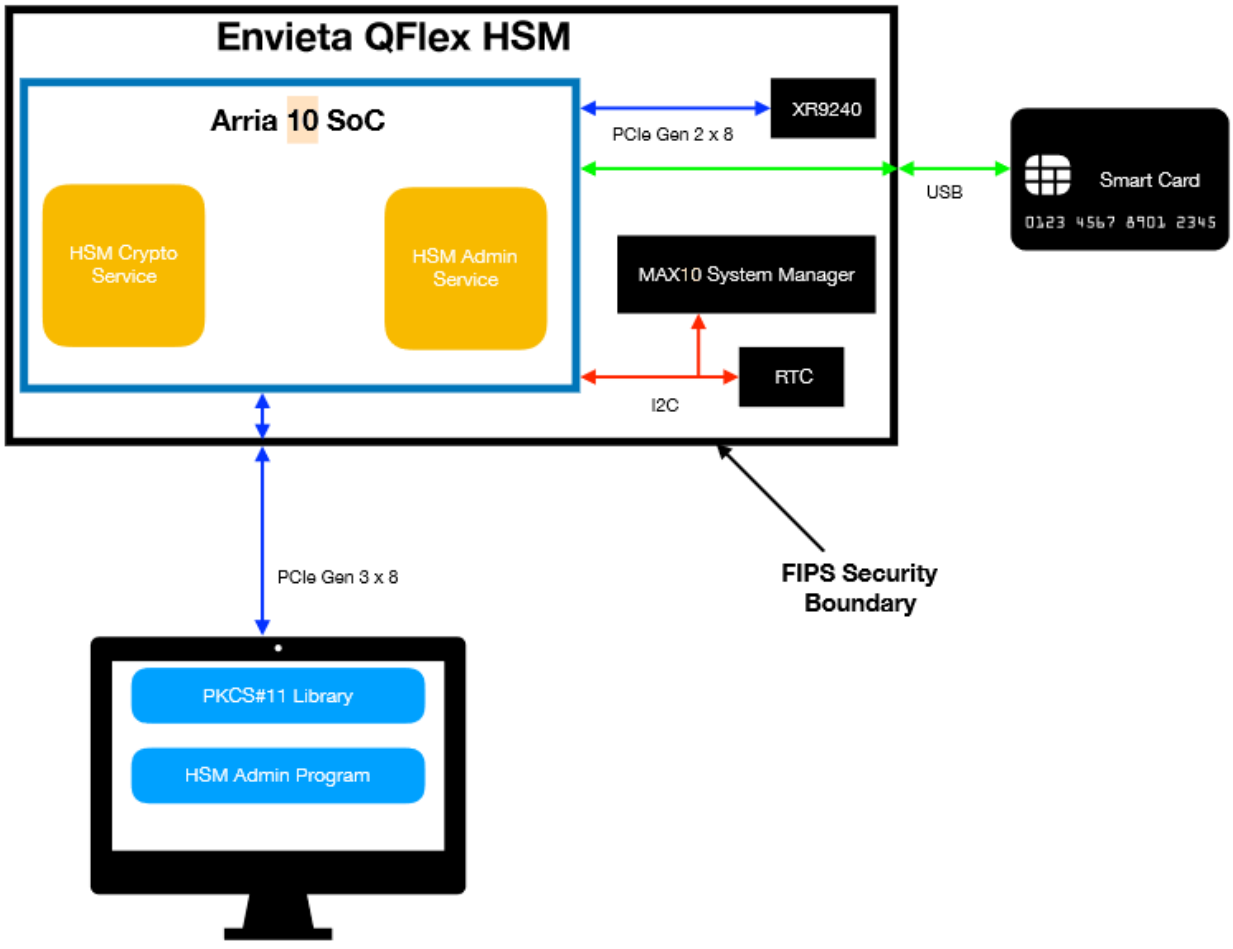

Figure 1 – QFlex HSM Cryptographic Boundary[1]

[1] Rev B models contain two instances of the RTC.

The physical boundary is the epoxy covering the entirety of the module's internal components as described in the previous section. The module is shown as assembled with a heatsink and a metal cover on the PCB depicted in Figure 2 below.

Figure 2 – QFlex HSM Physical Boundary – Rev A

Figure 3 - QFlex HSM Physical Boundary – Rev B

3.2 Physical HSM Components

The core of the Envieta QFlex HSM is the Arria 10 System-on-Chip (SoC). This SoC contains both an FPGA and a dual-core ARM Cortex-A9 hard-core processor. The FPGA fabric is used to support two PCIe engines; one for communicating to the host computer, and the other for communicating to the Exar XR9240 Security Co-processor and other fabric necessary for communicating with other peripherals on the HSM. The main software applications run on the ARM hard-processor core and interact with the rest of the system through the FPGA fabric.

The Exar XR9240 Data Compression and Security Coprocessor interfaces with the Arria 10 SoC through a PCIe Generation 2 bus of width 8 lanes. The security processor provides cryptographic acceleration for AES, RSA, ECDSA, Diffie-Hellman, digest, and message authentication operations. It also contains a nondeterministic random number generator.

The XR9240 comes packaged with a software library and driver that executes on the ARM Cortex-A9 hard core processor that is used by the HSM software components.

The MAX10 System Manager is used to control the power up sequence of the card and copies the FPGA bit stream out of QSPI flash memory.

The MCP79410 Real-Time-Clock is accessed via the I2C bus and contains a real-time-clock. Hardware version Rev A also contains 64-Bytes of Battery-Backed SRAM. The Battery-Backed SRAM contains the HSM Storage Derivation Key (HSDK).

Hardware version Rev B contains two Real-Time-Clocks, with two separate 64-Byte Battery-Backed SRAMs. One Battery-Backed SRAM contains the HSM Recoverable Tamper Key (RTK) and one contain the HSM Non-Recoverable Tamper Key (NRTK), which are used to obfuscate the HSM Storage Derivation Key (HSDK) for recovery purposes.

3.3 Logical HSM Components

The logical HSM components consist of two separate applications; the HSM Cryptographic Service Application (HCSA) and the HSM Administrative Service Application (HASA).

The HCSA supports messages that access the approved security functions of the HSM. These include, but are not limited to, performing key generation, encryption and decryption of data, message authentication, and signature generation and verification. The HCSA utilizes the XR9240 to perform accelerated cryptographic functions.

The HASA supports messages that perform maintenance operations on the HSM. Those operations include, but are not limited to, HSM backup and restore, HSM Slot backup and restore, log retrieval, and factory reset. The HASA utilizes the HCSA, via an internal communications protocol, to perform any required cryptographic functions.

4. Cryptographic Module Ports and Interfaces

4.1 Module Interface Description

The HSM has three physical interfaces:

- PCIe Gen3 8-lane connector
- Master USB connector where a Smart Card reader is connected
- Slave USB UART connector where status information is outputted
- Status LEDs

As required by FIPS 140-2, these interfaces are categorized according to one or more of the following:

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

The module's physical interfaces are mapped to the following logical interfaces supported by the QFlex HSM according to Table 2:

Logical Interface	Physical Interface
Data Input interface	PCIe, USB (Master)
Data Output interface	PCIe, USB (Master)
Control Input interface	PCIe, Pin Header (Rev B only), Push Button
Status Output interface	Status LED, PCIe, USB (Slave)

Table 2 – QFlex HSM Interfaces

The following logical interfaces will flow across the PCIe interface: the Data Input Interface, the Data Output Interface, the Control Input Interface, and the Status Output Interface. The PCIe interface is where the main command and control of the HSM will is performed. Over this physical port, there are three programmatic interfaces that are used by an operator to control the HSM: Cryptographic Maintenance Interface, Cryptographic Acceleration Interface, and Host Maintenance Interface. Through these interfaces data, control, and status information are transported. The messages are carried over the PCIe bus using an IP-over-PCIe protocol.

The following logical interfaces will flow across the USB master interface: the Data Input Interface, and the Data Output Interface. Over this interface, the HSM will command and control the attached Smart Card Reader and the Smart Card plugged into the reader. The Smart Card will be used to transport and securely store CSPs. Because of this, only the Data Input and Data Output logical interfaces are connected.

The USB slave interface is connected to a UART that shows console output from the HSM. The firmware only transmits data from this interface and does not process any input entering this interface. This interface only supports the Status output interface.

There is also a bank of LEDs that will be used as the Status Output Interface. The QFlex HSM features 4 LEDs which convey different modes and statuses as shows below.

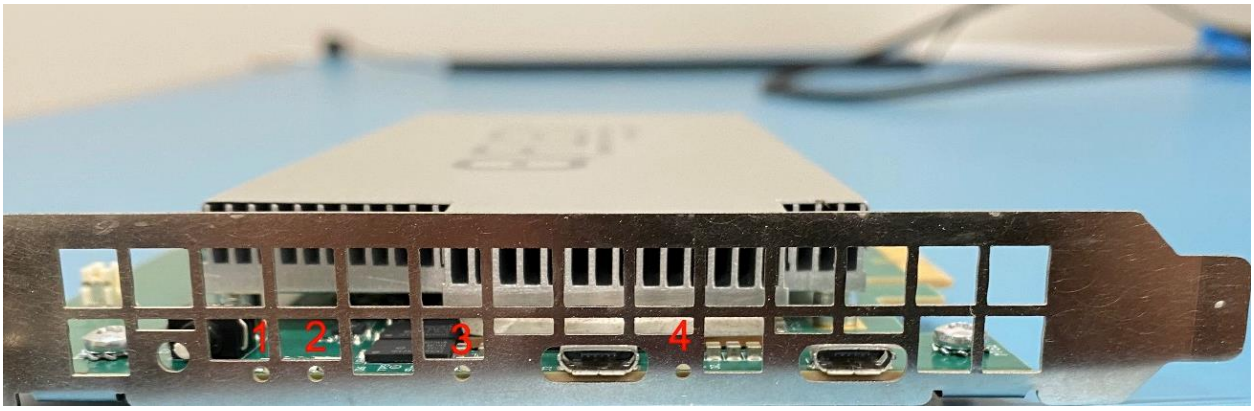

Figure 4 – Qflex HSM LED's

LEDs 1 & 2 convey specific hardware statuses for troubleshooting. These generally do not need to be monitored by the user.

LED 3 conveys the FIPS Status with the following color possibilities:

- Yellow -> Performing POSTs
- Cyan -> Factory Reset
- Green -> FIPS Mode
- Red -> FIPS Error Condition

LED 4 conveys HSM firmware status:

- Blinking Blue/White -> Normal Operating Mode
- Red -> Firmware Error
- Yellow -> Bootloader Stage
- Blinking Blue -> Secure Boot
- Solid Blue -> Kernel/Filesystem Loading

The Push Button interface triggers a factory reset operation, while the Header Pin is used to receive a signal from outside of the cryptographic boundary which will clear the Recoverable Tamper Key. This will place the module in the "RTK Missing State", where the CO can either recover the Recoverable Tamper Key or perform a factory reset.

5. Roles, Services and Authentication

5.1 Roles

The HSM supports the defined “User Role” and “Crypto Officer Role” using Identity based authentication. These roles are mapped to three operator types; two which are mapped to the Crypto Officer role and one which is associated with the user role.

- Slot User (User role)
- Slot Security Officer (Crypto Officer role)
- HSM Administrator (Crypto Officer role)

These roles represent a default grouping of permissions that enable the ability to execute the module’s services.

There is no Maintenance Role defined for the module.

5.1.1 User Role

The Slot User operates the HSM by utilizing the Cryptographic Maintenance Interface, and the Cryptographic Acceleration Interface. In this role, the user is allowed to access CSPs, generate and import new CSPs, export CSPs in encrypted form, and perform cryptographic operations. They may change their own password as well. The slot user is permitted to utilize the Host Maintenance Interface only for authenticating to the HSM via the use of smartcards.

The Slot User role includes by default the “Object Management”, “Object Export”, and “Cryptographic User” permissions.

5.1.2 Crypto Officer (CO) Role

There are two user types that fill the role of the Crypto Officer. First is the Slot Security Officer. The slot security officer can perform the same operations as the user but is also allowed to initialize and change a user’s password, and to initialize the token. This can be done using the Cryptographic Maintenance Interface and the Cryptographic Acceleration Interface. The slot security officer is also permitted to utilize the Host Maintenance Interface. Through that interface, the slot security officer is allowed to backup and restore the contents of the slot and retrieve logging information from the slot.

The Slot Security Officer by default includes the “User Management”, “Cloning”, “Object Management”, “Object Export”, and “Security Officer” permissions.

The second is the HSM Administrator. The HSM Administrator can perform HSM maintenance operations by utilizing the Host Maintenance Interface. Through this interface, the HSM Administrator is allowed to backup and restore the HSM, retrieve the HSM logs, set the time, initialize slot tokens, as well as configure permissions

The HSM Administration role includes the “HSM Administration”, “Firmware Administration”, “User Management”, “Cloning”, “Object Management”, “Object Export”, “Security Officer”, and “Cryptographic User” permissions.

5.2 Authentication

An operator of the HSM must authenticate themselves as one of these users in order to perform any cryptographic service that utilizes keys or to perform any actions that modify the state of the HSM.

An unauthenticated user is not allowed to perform any security related functionality and is only permitted to query the current FIPS state of the HSM, obtain random data and perform digest operations.

Role	Type of Authentication	Authentication Data
Slot User (User)	Identity Based	2048 Bit RSA Public Key
	Identity Based	Password
Slot Security Officer and HSM Administrator (Crypto Officer)	Identity Based	2048 Bit RSA Public Key
	Identity Based	Password

Table 3 – Roles and Authentication Data

HSM operators are authenticated using identity-based authentication. The user provides their username, which is either CKU_USER or CKU_SO when using the PKCS#11 API, or their respective username when using the administrative interface, and also authentication data. That authentication data may be supplied as a password or single smart card and PIN.

5.2.1 Password Authentication

Passwords are not sent in cleartext to the HSM. Instead, they are encrypted as part of using the Cryptographic Maintenance Interface or Host Maintenance Interface, depending upon who is logging in. Passwords are stored as salted digests in the HSM's non-volatile memory. On a login attempt, the salt is extracted from that user's password entry, added to the provided password, and then digested using the Exar XR9240. The type of digest used is SHA-512. If the digested value matches the value stored within the password database, the user is authenticated.

A password must be at least six (6), characters long. The acceptable character set is any upper or lower case letter, numbers, and the following special characters: ~`!@#$%^&*()_-+={}[]|\:;"'<>,./?. Any failed password attempts cause the thread attempting the authentication to pause for one (1) second.

The module supports a character set of 95 possible characters. At a minimum length of 6 characters, this puts the probability of being guessed at 1 in 95^6 (735,091,890,625), which is less than 1 in 1,000,000. At a rate of 1 attempt per second (the maximum rate at which login attempts can be accepted), the probability of guessing the password within a one-minute period becomes 1 in 95^6/60 (12,251,531,510), which is less than 1 in 100,000.

5.2.2 Single Smart Card Authentication

This method of authentication is available to all operators of the HSM. If a slot user attempts this method, it is initiated through the HCSA and completed using the HASA. When selected, a smart card is inserted into the smart card reader. The user is then required to enter a minimum 6-character alphanumeric PIN to unlock the smart card. When a smart card is first associated with a user, an RSA 2048-bit key pair is

generated on the smart card and the public key is extracted and stored in the HSM's password database on the HSM's nonvolatile memory. To verify the user, random data is generated using the XR9240 DRBG and passed to the smart card. The smart card then signs that data using the stored RSA private key and the signature is returned. The HSM then uses the stored RSA public key and previously generated random data and attempts to verify the signature using the XR9240. If the signature successfully verifies, the user is then authenticated.

With a 2048-bit key pair, the probability of randomly guessing the authentication credential is 1 in 2^112 which is less than 1 in 1,000,000. The probability of randomly guessing the authentication credential within a one-minute period is 1 in (2^112)/60 which is far less than 1 in 100,000. This probability is based on the module being able to process one authentication attempt per second, however, in practice the process takes well over a second per smart card read operation, resulting in an even lower probability.

5.3 Services

The HSM supports a number of services that are available to authenticated users. The module implements the following access control policy on keys and CSPs in the module shown in the following table. The access policy is noted by R=Read, W=Write, X=Execute, and Z = Zeroize.

Service	Permission Required	Role Required	Key and CSP Access
Change Password (Self)	Any Authenticated User	User, CO	Operator Password – W
Logout	Any Authenticated User	User, CO	N/A
Encrypt Data	Any Authenticated User	User, CO	User - Data Encryption Key – RX
Decrypt Data	Any Authenticated User	User, CO	User - Data Encryption Key – RX
Authenticated Encryption (GCM)	Any Authenticated User	User, CO	User - Data Encryption Key – RX
Authenticated Decryption (GCM)	Any Authenticated User	User, CO	User - Data Encryption Key – RX
Generate MAC (HMAC, GMAC, CMAC)	Any Authenticated User	User, CO	User - Data Authentication Key – RX
Verify MAC (HMAC, GMAC, CMAC)	Any Authenticated User	User, CO	User - Data Authentication Key – RX

Service	Permission Required	Role Required	Key and CSP Access
Generate Digital Signature	Any Authenticated User	User, CO	Private portion of User - RSA Key Pair – RX Private portion of User - ECDSA Key Pair – RX Private portion of User - DSA Key Pair – RX
Verify Digital Signature	Any Authenticated User	User, CO	Public portion of User - RSA Key Pair – RX Public portion of User - ECDSA Key Pair – RX Public portion of User - DSA Key Pair – RX
Import / Export Slot Backup Key	Cloning	CO	Slot Backup Key – RW Key Split Export Key Encryption Key - RX
Slot Backup / Restore	Cloning	CO	Slot Backup Key – RX
Import / Export HSM Backup Key	Cloning	CO	HSM Backup Key – RW Key Split Export Key Encryption Key – RX
HSM Backup / Restore	Cloning	CO	HSM Backup Key – RX
Initialize HSM	HSM Administration	CO	HSM Communications Key – W HSM Communications Certificate – W HSM Storage Derivation Key – W HSM Recoverable Tamper Key – RWX (Available in Rev B only) HSM Non-Recoverable Tamper Key – RWX (Available in Rev B only) HSM Communications Key Encryption Key – X HSM Device Integrity Key Encryption Key – X HSM Device Integrity Key – RX
Set HSM Clock	HSM Administration	CO	N/A
Zeroization (Factory Reset)	HSM Administration	CO	All CSPs – Z
Import / Export Recoverable Tamper Key	HSM Administration	CO	HSM Recoverable Tamper Key – RW (Available in Rev B only)
Smartcard Import / Export Recoverable Tamper Key	HSM Administration	CO	HSM Recoverable Tamper Key – RW (Available in Rev B only)
Perform Firmware Update	Firmware Administration	CO	QFlex PCIE FIPS Firmware Update Certificate – RX Envieta Root Certificate Authority – RX Firmware Update Certificate Authority – RX License Key Certificate Authority – RX HSM Device Integrity Certificate – RX All CSPs – Z (If changing firmware to a non-approved firmware load)
Load Firmware Licenses	Firmware Administration	CO	License Key Certificate Authority – RX Envieta Root Certificate Authority – RX
Generate Secret Key	Object Management	User, CO	User - Data Authentication Key – W DRBG Output – RX

Service	Permission Required	Role Required	Key and CSP Access
Generate Symmetric Key	Object Management	User, CO	User - Data Encryption Key – W DRBG Output – RX
Generate AES Key Wrapping Key	Object Management	User, CO	User - AES Key Wrapping Key – W DRBG Output – RX
Generate Key Pair / Domain Parameters	Object Management	User, CO	User - RSA Key Pair – W User - ECDSA Key Pair – W User - DH Key Pair – W User - DSA Key Pair / Domain Parameters – W DRBG Output – RX
Key Agreement	Object Management	User, CO	User - ECDSA Key Pair – RX User - DH Key Pair / Domain Parameters – RX User - Data Encryption Key – W User - Data Authentication Key – W
Key Derivation (user)	Object Management	User, CO	User - Key Derivation Key – RX User - KDF Output – W
Wrap Key	Object Export	User, CO	User - AES Key Wrapping Key – RX User – RSA Public Key - RX
Unwrap Key	Object Management	User, CO	User - AES Key Wrapping Key – RX User – RSA Private Key - RX
Import / Export User Key Split via Smart Card	Object Management, Object Export	CO	User - Data Encryption Key – RW User - Data Authentication Key – RW User - Key Derivation Key – RW User - KDF Output – RW User - ECDSA Private Key – RW User - DSA Private Key – RW User - DH Private Key – RW User - RSA Private Key – RW User - AES Key Wrapping Key – RW Key Split Export Key Encryption Key – RX
Import / Export User Key Split via Encrypted Buffer	Object Management, Object Export	CO	User - Data Encryption Key – RW User - Data Authentication Key – RW User - Key Derivation Key – RW User - KDF Output - RW User - ECDSA Private Key – RW User - DSA Private Key – RW User - DH Private Key – RW User - RSA Private Key – RW User - AES Key Wrapping Key - RW Key Split Export Key Encryption Key – RX User Password Data Protection Key – RX
Set User Password	Security Officer	CO	Operator Password – W

Service	Permission Required	Role Required	Key and CSP Access
Initialize	Unauthenticated[2]	N/A	Ephemeral Communications Key Pair – RX Ephemeral Communications Key – RWX
Get Info	Unauthenticated	N/A	N/A
Get FIPS Status	Unauthenticated	N/A	N/A
Get Logs	Unauthenticated	N/A	N/A
Perform Self-Tests	Unauthenticated	N/A	OS HMAC Signing Key – RX Root File System Signature Key – RX
Login (Password)	Unauthenticated	N/A	Operator Password – RX
Login (Single Smart Card)	Unauthenticated	N/A	Operator Authentication Public Key – RX
Generate SHA Digest	Unauthenticated	N/A	N/A
Generate Random Data	Unauthenticated	N/A	DRBG Output – RW
Seed RNG	Unauthenticated	N/A	Entropy Input String – RWX DRBG Seed – RWX DRBG Key – RWX DRBG V – RWX
Decommission	Unauthenticated	N/A	All CSPs – Z
Change Password	User Management	CO	Operator Password – W
Initiate Single-Smartcard Login	User Management	CO	Operator Authentication Public Key – W

Table 4 – Approved Roles, Services and CSP Access

[2] The Unauthenticated services as described are allowed per IG 3.1 exceptions a), b)

6. Physical Security

The module is a multiple-chip embedded cryptographic module made of production-grade materials. All integrated circuits are micro coated using industry-standard passivation techniques. Additionally, the PCB is coated with a hard epoxy potting material which covers all hardware and firmware components. The epoxy potting material is opaque and prevents physical access to any of the internal components such that attempts at accessing the internals is likely to result in destruction of the module.

7. Operational Environment

The operational environment of the module is non-modifiable and therefore the requirements of this section are not applicable. The operating environment of the module does not provide access to the underlying Linux kernel running on the Cortex A9 processor. Due to the way the system is booted, only signed images are executed on the processor. The main applications are contained in a read-only file system. The keys and other non-volatile items that are generated are written to a read/write area. All persistently stored CSPs are encrypted at rest.

8. Cryptographic Algorithms and Key Management

8.1 Cryptographic Algorithms

The module implements the following Approved algorithms. There are algorithms, modes, and keys that have been CAVs tested but not used by the module. Only the algorithms, modes/methods, and key lengths/curves/moduli shown in this table are used by the module

CAVP Cert #	Algorithm	Standard	Mode/ Method	Key Lengths, Curves, or Moduli	Use
2958	AES	FIPS PUB 197 NIST SP 800-38A	CBC CTR ECB	128 192 256	Encryption/Decryption
		FIPS PUB 197 NIST SP 800-38D	GCM[3]	128 192 256	Authenticated Encryption/Decryption
1877	HMAC	FIPS PUB 198-1	SHA-1 SHA2-224 SHA2-256 SHA2-384 SHA2-512	160 bits 224 bits 256 bits 384 bits 512 bits	Keyed-Hash Message Authentication
2490	SHS	FIPS PUB 180-4	SHA-1 SHA2-224 SHA2-256 SHA2-384 SHA2-512	N/A	Hashing
559	DRBG	SP 800-90A	CTR	128 192 256	Random Bit Generation

Table 5 – Approved Algorithms (Exar Chip)

CAVP Cert #	Algorithm	Standard	Mode/ Method	Key Lengths, Curves, or Moduli	Use
N/A[4]	CKG[5]	SP 800-133 rev 2			Cryptographic Key Generation
C893	AES	FIPS PUB 197 NIST SP 800-38B	CMAC	128 192 256	Message Authentication
N/A	KTS	FIPS PUB 197 NIST SP 800-38F NIST SP 800-38D	AES Cert. #C893, KW, KWP, GCM		Key Wrapping; Key establishment methodology provides between 128 and 256 bits of encryption strength

[3] The module meets scenario 2 of IG A.5. The IV is at least 96-bits in length as generated in its entirety internally by the module's Approved DRBG.
[4] Vendor affirmed
[5] The CSPs output from the DRBG are constructed using Scenario 1 of Section 4 in SP 800-133rev2.

CAVP Cert #	Algorithm	Standard	Mode/ Method	Key Lengths, Curves, or Moduli	Use
C893	RSA	FIPS PUB 186-4	PKCS1.5 PSS	2048 bit: SHA2-224 SHA2-256 SHA2-384 SHA2-512 3072 bit: SHA2-224 SHA2-256 SHA2-384 SHA2-512 4096 bit[6]: SHA2-224 SHA2-256 SHA2-384 SHA2-512 6144 bit: SHA2-224 SHA2-256 SHA2-384 SHA2-512	Signature Generation

[6] RSA 4096 and 6144 SigGen / SigVer were not tested by the CAVP; however, they are Approved for use per IG A.14, because RSA SigGen / SigVer were tested in accordance with FIPS 186-4 for the 2048 and 3072 bit modulus sizes, and testing for modulus sizes higher than 3072 is not available under CAVS.

CAVP Cert #	Algorithm	Standard	Mode/ Method	Key Lengths, Curves, or Moduli	Use
			PKCS1.5 PSS	1024-bit: SHA-1 SHA-224 SHA-256 SHA-384 SHA-512 2048 bit: SHA-1 SHA2-224 SHA2-256 SHA2-384 SHA2-512 3072 bit: SHA-1 SHA2-224 SHA2-256 SHA2-384 SHA2-512 4096 bit: SHA-1 SHA2-224 SHA2-256 SHA2-384 SHA2-512 6144 bit: SHA-1 SHA2-224 SHA2-256 SHA2-384 SHA2-512	Signature Verification
C893	ECDSA	FIPS PUB 186-4	KeyGen	P-224 P-256 P-384 P-521	Key Generation
			KeyVer	P-192 P-224 P-256 P-384 P-521	Key Verification

CAVP Cert #	Algorithm	Standard	Mode/ Method	Key Lengths, Curves, or Moduli	Use
			SigGen	P-224: SHA2-224 SHA2-256 SHA2-384 SHA2-512 P-256: SHA2-224 SHA2-256 SHA2-384 SHA2-512 P-384: SHA2-224 SHA2-256 SHA2-384 SHA2-512 P-521: SHA2-224 SHA2-256 SHA2-384 SHA2-512	Signature Generation

CAVP Cert #	Algorithm	Standard	Mode/ Method	Key Lengths, Curves, or Moduli	Use
			SigVer	P-192: SHA-1 SHA2-224 SHA2-256 SHA2-384 SHA2-512 P-224: SHA-1 SHA2-224 SHA2-256 SHA2-384 SHA2-512 P-256: SHA-1 SHA2-224 SHA2-256 SHA2-384 SHA2-512 P-384: SHA-1 SHA2-224 SHA2-256 SHA2-384 SHA2-512 P-521: SHA-1 SHA2-224 SHA2-256 SHA2-384 SHA2-512	Signature Verification

CAVP Cert #	Algorithm	Standard	Mode/ Method	Key Lengths, Curves, or Moduli	Use
C893	DSA	FIPS PUB 186-4	PQGGen	L = 2048, N=224: SHA2-224 SHA2-256 SHA2-384 SHA2-512 L=2048, N=256: SHA2-256 SHA2-384 SHA2-512 L=3072, N=256: SHA2-256 SHA2-384 SHA2-512	DSA Parameter Generation
			PQGVer	L = 2048, N=224: SHA2-224 SHA2-256 SHA2-384 SHA2-512 L=2048, N=256: SHA2-256 SHA2-384 SHA2-512 L=3072, N=256: SHA2-256 SHA2-384 SHA2-512	DSA Parameter Verification
			KeyGen	L=2048, N=224 L=2048, N=256 L=3072, N=256	Key Generation

CAVP Cert #	Algorithm	Standard	Mode/ Method	Key Lengths, Curves, or Moduli	Use
			SigGen	L = 2048, N=224: SHA2-224 SHA2-256 SHA2-384 SHA2-512 L=2048, N=256: SHA2-224 SHA2-256 SHA2-384 SHA2-512 L=3072, N=256: SHA2-224 SHA2-256 SHA2-384 SHA2-512	Signature Generation

CAVP Cert #	Algorithm	Standard	Mode/ Method	Key Lengths, Curves, or Moduli	Use
			SigVer	L = 1024, N=160: SHA-1 SHA2-224 SHA2-256 SHA2-384 SHA2-512 L = 2048, N=224: SHA-1 SHA2-224 SHA2-256 SHA2-384 SHA2-512 L=2048, N=256: SHA-1 SHA2-224 SHA2-256 SHA2-384 SHA2-512 L=3072, N=256: SHA-1 SHA2-224 SHA2-256 SHA2-384 SHA2-512	Signature verification
N/A[7]	KAS-SSC	SP 800-56A rev 3	ECC (Cofactor) Ephemeral Unified ECC (Cofactor)_ Full Unified Model	P-521	Key Agreement (Shared secret computation) with a Key Derivation as specified in SP 800-56C rev 1. Key establishment methodology provides 256 bits of encryption strength.

[7] Vendor-affirmed

CAVP Cert #	Algorithm	Standard	Mode/ Method	Key Lengths, Curves, or Moduli	Use
N/A[8]	KAS-SSC	SP 800-56A rev 3	FFC dhEphem,	ffdhe2048 ffdhe3072 ffdhe4096 ffdhe6144 ffdhe8192	Key Agreement (Shared secret computation) with a Key Derivation as specified in SP 800-56C rev 1. Key establishment methodology provides between 112 and 202 bits of encryption strength
N/A[9]	KDA	800-56C rev 1	One Step Key Derivation Option 1	SHA2-224 SHA2-256 SHA2-384 SHA2-512	Key Derivation
C893	KBKDF	SP 800-108	Counter Mode CMAC-AES128 CMAC-AES192 CMAC-AES256 HMAC SHA-1 HMAC SHA2-224 HMAC SHA2-256 HMAC SHA2-384 HMAC SHA2-512	Between 14 bytes and 1024 bytes	Key Derivation
N/A[10]	PBKDF	SP 800-132	HMAC-SHA-1 HMAC-SHA-224 HMAC-SHA-256 HMAC-SHA-384 HMAC-SHA-512	Between 14 bytes and 1,024 bytes.	Key Derivation SP800-132: Option 1a

Table 6 – Approved Algorithms (Qflex HSM Cryptographic Library)

[8] Vendor-affirmed
[9] Vendor-affirmed
[10] Vendor-affirmed

CAVP Cert #	Algorithm	Standard	Mode/ Method	Key Lengths, Curves, or Moduli	Use
C892	RSA	FIPS PUB 186-4	PKCS1.5 PSS	2048 bit: SHA2-256 SHA2-512 3072 bit: SHA2-256 SHA2-512	Signature Verification
C892	HMAC	FIPS PUB 198-1	SHA2-256 SHA2-512	256 bits 512 bits	Keyed-Hash Message Authentication
C892	SHS	FIPS PUB 180-4	SHA2-256 SHA2-512	N/A	Hashing

Table 7 – Approved Algorithms (Qflex HSM Secure Boot Library)

CAVP Cert #	Algorithm	Standard	Mode/ Method	Key Lengths, Curves, or Moduli	Use
A1471	RSA	FIPS PUB 186-4	X9.31	2048 bit 3072 bit 4096 bit 6144 bit[11]	Key Generation
A1471	KTS-RSA	SP800-56Br2	KTS-OAEP-basic Initiator, Responder, SHA2-224, SHA2-256, SHA2-384, SHA2-512	2048-bit 3072-bit 4096-bit 6144-bit	Key Wrapping; key establishment methodology provides between 112 and 178 bits of encryption strength

Table 8 – Approved Algorithms (QFlex HSM RSA Key Generation and Transport Cryptographic Library)

8.1.1 Non-Approved Algorithms

The following algorithms are non-approved but allowed:

- NDRNG used for seeding the Approved CTR-DRBG. The NDRNG provides a minimum of 0.73 bits of entropy per 1-bit sample.

[11] RSA 6144 bit KeyGen was not tested by the CAVP; however, it is Approved for use per IG A.14, because RSA KeyGen was tested in accordance with FIPS 186-4 with the 2048, 3072 and 4096 bit modulus sizes, and testing for the higher modulii was not available. The generation of random probable primes is performed in accordance with Section B.3.3.

8.1.2 Vendor Documentation for SP800-132 – Password Based Key Derivation

Keys derived using PBKDF2 shall only be used in storage applications.

The minimum password length allowed is 6, alpha-numeric characters. This puts the probability of the password being guessed at 1 in 735,091,890,625. A larger, more complex password is recommended to further decrease the probability of the password being guessed.

The minimum number of iterations that can be performed is 1,000. The minimum salt length is 128 bits.

8.2 Cryptographic Keys and CSPs

The cryptographic keys and CSPs used by the module are described in Table 9:

Keys / CSP	Description	Key / CSP Type	Generation/ Input	Output	Storage	Zeroization
Operator Passwords	Used for Authentication	Password must be at least six (6), valid characters using any lower-case or capital letter, number, or special character as defined in Section 5.2.1.	Entered from user via approved KTS (IG D.9)	If using multi-smartcard login, output via approved KTS (IG D.9) to smart cards	Password not directly stored, only the salted, SHA512 digest of it (non-volatile memory)	When overwritten with new password or during factory reset
Operator Authentication Public Key	Used to verify a smart card for single smart card login	2048-bit RSA public key	Generated on a smart card and entered into the device via the USB attached card reader (IG 2.1)	N/A	Non-volatile Memory (plaintext)	When the user changes the smart card to be used, when manually deleted, or during factory reset
Key Split Export Key Encryption Key	Used to encrypt a CSP that is exported from the module using split knowledge.	AES-GCM 256-bit key	Generated using the DRBG and SP800-133rev2 Section 4, Scenario 1 Input via split knowledge (IG 2.1)	Output via split knowledge (IG 2.1)	Volatile memory (plaintext)	Zeroized upon power cycle

Keys / CSP	Description	Key / CSP Type	Generation/ Input	Output	Storage	Zeroization
User Password Data Protection Key	Used to wrap the key-split blob that exits the module via split knowledge (IG 2.1). Key-split blobs may contain: • "User" CSPs • HSM/Slot Backup Key	AES-GCM 256-bit key	Generated using SP800-132, option 1a.	N/A	Volatile memory (plaintext)	Zeroized upon power cycle
HSM Storage Derivation Key (Rev A)	Used to derive the Slot Encryption Keys and AES Key Wrapping Keys	32-byte generic secret key	Generated using the DRBG and SP800-133rev2 Section 4, scenario 1	N/A	Battery-backed memory (plaintext)	Zeroized during factory reset or if battery powering battery-backed memory is removed
HSM Storage Derivation Key (Rev B)	Used to derive the Slot Encryption Keys and AES Key Wrapping Keys	32-byte generic secret key	Generated using the DRBG and SP800-133rev2 Section 4, scenario 1	N/A	Volatile memory (plaintext)	Zeroized upon power cycle
HSM Non-Recoverable Tamper Key (Rev B)	Used with HSM Recoverable Tamper Key to generate HSM Storage Derivation Key.	32-byte generic secret key	Generated using the DRBG and SP800-133rev2 Section 4, scenario 1	N/A	Battery-backed memory (plaintext)	Zeroized during factory reset or if battery powering battery-backed memory is removed

Keys / CSP	Description	Key / CSP Type	Generation/ Input	Output	Storage	Zeroization
HSM Recoverable Tamper Key (Rev B)	Used with HSM Non-Recoverable Tamper Key to recover HSM Storage Derivation Key. This is the resultant value when the two values are XORed together (IG 1.23 Scenario 1).	32-byte generic secret key	Created by obfuscating the HSM Storage Derivation Key by XORing it with the Non-Recoverable Tamper Key. Input via split knowledge (IG 2.1)	Output via split knowledge (IG 2.1)	Battery-backed memory (plaintext)	Zeroized during factory reset, if battery powering battery-backed memory is removed, or when signal is received from outside the cryptographic module boundary on the Pin Header interface.
Slot Encryption Key	Encryption of secret and private keys within non-volatile memory	AES-GCM 256-bit key	Derived from HSM Storage Derivation Key according to SP 800-108 KDF	N/A	Volatile memory (plaintext)	Zeroized upon power cycle
HSM / Slot Backup Key	Encryption of the contents of HSM / slot backup	AES-GCM 256 bit key	Generated using the DRBG and SP800-133rev2 Section 4, scenario 1 Input via split knowledge (IG 2.1)	Output via split knowledge (IG 2.1)	Non-volatile memory (encrypted) Volatile memory during operation (plaintext)	Keys in non-volatile memory zeroized during factory reset Keys in volatile memory are zeroized upon power cycle

Keys / CSP	Description	Key / CSP Type	Generation/ Input	Output	Storage	Zeroization
User - Data Encryption Key	Symmetric encryption services	AES-CBC, CTR, ECB, GCM 128, 192, and 256-bit keys	Generated using the DRBG and SP800-133rev2 Section 4, scenario 1 Input via approved KTS (IG D.9), or split knowledge (IG 2.1)	Output via approved KTS (IG D.9) or split knowledge (IG 2.1)	Non-volatile memory (encrypted) Volatile memory during operation (plaintext)	Keys in non-volatile memory zeroized during factory reset Keys in volatile memory are zeroized upon power cycle
User - Data Authentication Key	Hash-based MAC services	HMAC-SHA-1, HMAC-SHA2-256, HMAC-SHA2-384, HMAC-SHA-512 key AES-GMAC, CMAC 128, 192, 256-bit key	Generated using the DRBG and SP800-133rev2 Section 4, scenario 1 Input via approved KTS (IG D.9) or split knowledge (IG 2.1)	Output via approved KTS (IG D.9) or split knowledge (IG 2.1)	Non-volatile memory (encrypted) Volatile memory during operation (plaintext)	Keys in non-volatile memory zeroized during factory reset Keys in volatile memory are zeroized upon power cycle
User – Key Derivation Key	Key used as input to an approved KBKDF or Concat KDF	Generic key of variable size. AES 128, 192, 256-bit key for KBKDF when using AES-CMAC for PRF	Generated using the DRBG and SP800-133rev2 Section 4, scenario 1 Input via approved KTS (IG D.9) or split knowledge (IG 2.1)	Output via approved KTS (IG D.9) or split knowledge (IG 2.1)	Non-volatile memory (encrypted) Volatile memory during operation (plaintext)	Keys in non-volatile memory zeroized during factory reset Keys in volatile memory are zeroized upon power cycle

Keys / CSP	Description	Key / CSP Type	Generation/ Input	Output	Storage	Zeroization
User – KDF Output	Derived key from KBKDF or Concat KDF function	Generic key of variable size AES 128, 192, 256-bit key	Derived according to SP 800-108 or SP 800-56C Section 4.1	Output via approved KTS (IG D.9) or split knowledge (IG 2.1)	Non-volatile memory (encrypted) Volatile memory during operation (plaintext)	Keys in non-volatile memory zeroized during factory reset Keys in volatile memory are zeroized upon power cycle
User – AES Key Wrapping Key	Transporting secret and private keys being entered into or output from the module.	AES GCM, KW or KWP 128, 192, or 256 bit keys	Generated using the DRBG and SP800-133rev2 Section 4, scenario 1 Input via approved KTS (IG D.9) or split knowledge (IG 2.1)	Output via approved KTS (IG D.9) or split knowledge (IG 2.1)	Non-volatile memory (encrypted) Volatile memory during operation (plaintext)	Keys in non-volatile memory zeroized during factory reset Keys in volatile memory are zeroized upon power cycle
User – RSA Public Key Wrapping Key	Transporting secret and private keys being output from the device	2048-bit 3072-bit 4096-bit 6144-bit	Generated using the DRBG and SP800-133 rev 2 Section 4, scenario 1 and FIPS PUB 186-4 Public keys Input via approved KTS (IG D.9)	Public Keys output via approved KTS (IG D.9).	Non-volatile memory (encrypted) Volatile memory during operation (plaintext)	Keys in non-volatile memory zeroized during factory reset Keys in volatile memory are zeroized upon power cycle

Keys / CSP	Description	Key / CSP Type	Generation/ Input	Output	Storage	Zeroization
User – RSA Private Key Unwrapping	Transporting secret and private keys being input into the device	2048-bit 3072-bit 4096-bit 6144-bit	Generated using the DRBG and SP800-133 rev 2 Section 4, scenario 1 and FIPS PUB 186-4 Private keys input via approved KTS (IG D.9) or split knowledge (IG 2.1).	Private Keys output via approved KTS (IG D.9) or split knowledge (IG 2.1).	Non-volatile memory (encrypted) Volatile memory during operation (plaintext)	Keys in non-volatile memory zeroized during factory reset Keys in volatile memory are zeroized upon power cycle
Ephemeral Communications Key Pair / Domain Parameters	Establishment of Ephemeral Communications Key	ECC CDH P-521 key pair	Generated according to SP 800-56A and FIPS PUB 186-4	N/A	Volatile memory	Zeroized at completion of key exchange
Ephemeral Communications Key	Protection of communication between operators and module	AES-GCM 256-bit key	Established according to SP 800-56A rev3 C(2e,0s, ECCCDH) and SP800-56C Section 4.1	N/A	Volatile memory	Zeroized when session completes

Keys / CSP	Description	Key / CSP Type	Generation/ Input	Output	Storage	Zeroization
User - DSA Key Pair / Domain Parameters	Signature services	[L, N]: [2048, 224] [2048, 256] [3072, 256]	Generated using the DRBG and SP800-133rev2 Section 4, scenario 1 and FIPS PUB 186-4 Public keys and domain parameters Input via approved KTS (IG D.9). Private keys input via approved KTS (IG D.9) or split knowledge (IG 2.1).	Public Keys and domain parameters output via approved KTS (IG D.9). Private Keys output via approved KTS, (IG D.9) or split knowledge (IG 2.1).	Non-volatile memory (encrypted) Volatile memory during operation (plaintext)	Keys in non-volatile memory zeroized during factory reset Keys in volatile memory are zeroized upon power cycle
User - ECDSA Key Pair	Signature services, ECC CDH key agreement	P-224, P-256, P-384, P-521	Generated using the DRBG and SP800-133rev2 Section 4, scenario 1 and FIPS PUB 186-4 Public keys Input via approved KTS (IG D.9). Private keys input via approved KTS (IG D.9) or split knowledge (IG 2.1).	Public Keys output via approved KTS. Private Keys output via approved KTS (IG D.9) or split knowledge (IG 2.1).	Non-volatile memory (encrypted) Volatile memory during operation (plaintext)	Keys in non-volatile memory zeroized during factory reset Keys in volatile memory are zeroized upon power cycle

Keys / CSP	Description	Key / CSP Type	Generation/ Input	Output	Storage	Zeroization
User – DH Key pair / Domain Parameters	FFC key agreement	ffdhe2048 ffdhe3072 ffdhe4096 ffdhe6144 ffdhe8192	Generated using the DRBG and SP800-133rev2 Section 4, scenario 1 and SP 800-56A rev 3 Public keys and domain parameters Input via approved KTS (IG D.9). Private keys input via approved KTS (IG D.9) or split knowledge (IG 2.1).	Public Keys and domain parameters output via approved KTS (IG D.9). Private Keys output via approved KTS (IG D.9) or split knowledge (IG 2.1).	Non-volatile memory (encrypted) Volatile memory during operation (plaintext)	Keys in non-volatile memory zeroized during factory reset Keys in volatile memory are zeroized upon power cycle
User - RSA Key Pair	Signature services	2048-bit, 3072-bit, 4096-bit, 6144-bit	Generated using the DRBG and SP800-133 rev 2 Section 4, scenario 1 and FIPS PUB 186-4 Public keys Input via approved KTS (IG D.9). Private keys input via encrypted channel, approved KTS (IG D.9) or split knowledge (IG 2.1).	Public Keys output via approved KTS (IG D.9). Private Keys output via approved KTS (IG D.9) or split knowledge (IG 2.1).	Non-volatile memory (encrypted) Volatile memory during operation (plaintext)	Keys in non-volatile memory zeroized during factory reset Keys in volatile memory are zeroized upon power cycle

Keys / CSP	Description	Key / CSP Type	Generation/ Input	Output	Storage	Zeroization
DRBG Entropy Input String	Random bit generation	DRBG input	Internally generated from hardware entropy source	N/A	Volatile memory	Keys in volatile memory are zeroized upon power cycle
DRBG Seed	Random bit generation	DRBG input	Internally generated from hardware entropy source	N/A	Volatile memory	Keys in volatile memory are zeroized upon power cycle
DRBG V	Random bit generation	Internal state value	Internal value used as part of SP 800-90a CTR_DRBG	N/A	Volatile memory	Keys in volatile memory are zeroized upon power cycle
DRBG Key	Random bit generation	Internal state value	Internal value used as part of SP 800-90a CTR_DRBG	N/A	Volatile memory	Keys in volatile memory are zeroized upon power cycle
Envieta Root Certificate Authority	Root certificate that is the root of trust for keys generated external to the HSM	X.509 Certificate with RSA 4096-bit Public key	Loaded at manufacturing	N/A	Non-volatile memory (plaintext within Signed Root File System)	N/A
Factory Integrity Certificate Authority	Certificate, signed by the Envieta Root Key, that authorizes a factory to produce HSMs and to validate HSMs	X.509 Certificate with RSA 4096-bit Public key	Loaded at manufacturing	N/A	Non-volatile memory (plaintext within Signed Root File System)	N/A
Production Factory Authorization Certificate Authority	Certificate authority for particular factory used to validate HSM Device Integrity Keys	X.509 Certificate with RSA 3072-bit Public key	Loaded at manufacturing	N/A	R/W memory (plaintext)	N/A

Keys / CSP	Description	Key / CSP Type	Generation/ Input	Output	Storage	Zeroization
HSM Device Integrity Key	Key used to sign certificates of other device key pairs generated on the HSM.	ECDSA P-384 Private Key	Generated on HSM at manufacturing	N/A	R/W memory (encrypted)	N/A
HSM Device Integrity Certificate	Public half of HSM Device Integrity Key. Signed by the Production Factory Integrity Key.	ECDSA P-384 Public Key contained in X.509 Certificate	Loaded at manufacturing	Public key output in plaintext	R/W memory (plaintext)	N/A
HSM Device Integrity Key Encryption Key	Key used to encrypt the HSM Device Integrity Key	AES 256-bit key	Derived from CO password using PBKDF and from HSDK KBKDF	N/A	Volatile memory (plaintext)	Keys in volatile memory are zeroized upon power cycle
Firmware Update Certificate Authority	Certificate that can be used to authenticate firmware update signatures. Signed by Envieta Root Key	X.509 Certificate with RSA 4096-bit Public key	Loaded at manufacturing	N/A	Non-volatile memory (plaintext within Signed Root File System)	N/A
QFlex PCIE Firmware Update Certificate Authority	Certificate that can be used to authenticate firmware update signatures for QFlex PCIe devices. Signed by the Firmware Update CA Key	X.509 Certificate with RSA 3072-bit Public key	Loaded at manufacturing	N/A	Non-volatile memory (plaintext within Signed Root File System)	N/A

Keys / CSP	Description	Key / CSP Type	Generation/ Input	Output	Storage	Zeroization
QFlex PCIE FIPS Firmware Update Certificate	Certificate used to authenticate FIPS firmware updates. Signed by QFlex PCIE Firmware Update CA	X.509 Certificate with RSA 3072-bit Public key	Loaded at manufacturing	N/A	Non-volatile memory (plaintext within Signed Root File System)	N/A
License Key Certificate Authority	Used to authenticate license signing certificates	X.509 Certificate with RSA 4096-bit Public key	Loaded at manufacturing	N/A	Non-volatile memory (plaintext within Signed Root File System)	N/A
OS HMAC Signing Key	Used to authenticate Linux Kernel image	32-byte HMAC Key	Loaded at manufacturing	N/A	R/O Flash (plaintext)	N/A
Root File System Signature Key	Used to authenticate root file system	RSA 4096-bit public Key	Loaded at manufacturing	N/A	R/O Flash (plaintext)	N/A
HSM Communications Key	Unused; reserved for future application	ECDSA P-521 Private key	Generated using the DRBG and SP800-133 rev 2 Section 4, scenario 1 and FIPS PUB 186-4	N/A	Non-volatile memory (encrypted)	Keys in non-volatile memory zeroized during factory reset
HSM Communications Certificate	Unused; reserved for future application	X.509 Certificate with ECDSA P-521 Public key	Generated using the DRBG and SP800-133 rev 2 Section 4, scenario 1 and FIPS PUB 186-4	N/A	Non-volatile memory (plaintext)	Keys in non-volatile memory zeroized during factory reset
HSM Communications Key Encryption Key	Unused; reserved for future application	AES 256-bit key	Derived from HSM Storage Derivation Key according to SP 800-108 KDF	N/A	Volatile memory (plaintext)	Keys in volatile memory are zeroized upon power cycle

Table 9 – Approved Keys and CSPs Table

8.3 Cryptographic Key Zeroization

Persistently stored keys are zeroized within the HSM when a factory reset operation is completed. Zeroization is handled differently between hardware versions Rev A and Rev B. Zeroization is primarily accomplished by deleting the HSM Storage Derivation Key. The HSM Storage Derivation Key serves as the root key to derive the keys used to decrypt slot key data, and with the root key material zeroized, sensitive data can no longer be decrypted and is no longer accessible for use by the HSM.

In version Rev A, when the HSM receives a factory reset command, or if constant power is removed from battery-backed memory, the HSM Storage Derivation Key is zeroized, the contents of the slot key databases are deleted, and the HSM is rebooted.

In version Rev B, the HSM Storage Derivation Key is cleared at every power cycle. The HSM Non-Recoverable Tamper Key is cleared when the HSM receives a factory reset command, or constant power is removed from its specific battery-backed memory. The HSM Recoverable Tamper Key is cleared when the HSM receives a factory reset command, or constant power is removed from its specific battery-backed memory. The HSM Recoverable Tamper Key's battery-backed memory is also cleared when a signal from outside the cryptographic boundary is received on the Pin Header interface.. This allows the module to enter a tamper state if the host computer it resides in is under maintenance.

In both revisions, ephemeral keys are zeroized at the completion of a user session, or upon a power cycle.

9. Self-Tests

FIPS 140-2 requires the module to perform self-tests to ensure the module integrity and the correctness of the cryptographic functionality at start up. Some functions also require conditional tests during normal operation of the module.

If any of the tests fail, the module enters an error state where no cryptographic functions can be executed (with the exception of the conditional Firmware Load Test). In order to attempt to clear the error, the operator must reboot or power cycle the module.

9.1 Power-On Self-Tests

Power-on self-tests are run upon the initialization of the module and do not require operator intervention to run. If any of the tests fail, the module will not initialize. The module will enter an error state and no cryptographic services can be accessed by the operator; the error state can be cleared by rebooting the module. If the error condition is not cleared, then the module is considered to be malfunctioning and should be returned to the manufacturer.

The module implements the following integrity tests:

- HMAC-SHA2-256 Kernel Integrity Test
- RSA-4096 with SHA-512 File System Integrity Test

The module implements the following power-on self-tests:

- AES-GCM KAT (encrypt – 256bits)
- AES-GCM KAT (decrypt – 256bits)
- AES-ECB KAT (encrypt)
- AES-ECB KAT (decrypt)
- SHA-1 KAT
- SHA-256 KAT
- SHA-512 KAT
- HMAC-SHA-256 KAT
- DRBG KAT
    - SP 800-90A Health Tests
- SP800-56A KAT for ECC
    - ECC CDH primitive KAT (Curve used: P-521)
- SP800-56A KAT for FFC
    - FFC DH primitive KAT (Key Size 4096 bits)
- Concat KDF (Sp800-56C 4.1 option 1) KAT
    - Utilizes SHA-256 as PRF
- KBKDF in counter mode using SHA-256 as PRF KAT
- DSA 2048/256 Sign/Verify with fixed keys PWCT
- ECDSA P-256 Sign/Verify with fixed keys PWCT
- RSA 2048-bit modulus using PKCS#1.5 padding Sign/Verify using fixed keys KAT
- SP800-56Br2 RSA Encryption and RSA Decryption Primitives

9.2 Conditional Self-Tests

Conditional self-tests are test that run during operation of the module. The module performs the following conditional self-tests:

Type	Test Description
Pairwise-consistency Test	Whenever an RSA, DSA or ECDSA key pair of any valid size is generated on the HSM, before the operation is completed and the keys are made available for use to the operator, a pair-wise consistency test is executed on the key pair.
Continuous Random Number Generator Test	Whenever random numbers are generated on the HSM Exar Non-deterministic RNG, the generated numbers undergo a continuous random number generator test. In accordance with IG 9.8, the CRNGT is not performed on the DRBG output, as SP800-90A assurances are executed after every 128KB of data is retrieved from the DRBG.
Firmware Load Test	When firmware is updated on the HSM, the update image must be validated before the underlying firmware on the device is updated. This is accomplished through an RSA-3072, SHA-256 RSA-PSS padded signature validation on the update image.

SP800-56A rev 3 Assurances	These assurances are performed whenever the SP 800-56A rev 3 key agreement scheme is executed. This includes checking the parameters, the received public key, and performing a key-pair PWCT when the ephemeral key pair is generated.

Table 10 – Conditional Self-tests

9.3 Critical Function Test

The first critical function test verifies the existence of the HSM Device Integrity Key, the HSM Device Integrity Certificate, the Production Factory Integrity Certificate, and the other Certificate Authorities described above in Table 9. If any of them are missing, the module enters the POST Error condition state. The only way to clear this error is to return the module to the manufacturer.

The second critical function test validates the existence of the HSM Storage Derivation Key (Rev A) or the Non-Recoverable Tamper Key and Recoverable Tamper Key (Rev B).

- On a Rev A module, the second critical function test verifies the HSM Storage Derivation Key stored in the HSM's real-time clock's (RTC) Battery-Backed RAM (BBR). A SHA-256 digest is calculated, and the calculated digest is compared against a previously calculated digest that is stored on the HSM after the POST completes. If the digests do not match, then the HSDK is deemed invalid, and the module is put into the factory reset state, in which all user data is cleared and the HSDK, HSM Communications Key, and HSM Communications Certificate are erased.
- On a Rev B module, two SHA-256 digests are calculated, one on the Non-Recoverable Tamper Key and one on the Recoverable Tamper Key. The Non-Recoverable Tamper Key digest is calculated and compared with the stored value. If not equal, the module goes into the factory reset state. Next, the Recoverable Tamper Key digest is calculated and compared with the stored value. If not equal, the module enters a "RTK Missing State" (only present on a Rev B module), where the CO can restore the Recoverable Tamper Key from a smartcard or encrypted file, or perform a factory reset. Finally, the HSDK checksum is computed as it would on a Rev A module. If not equal to the stored value, the module enters a factory reset.

The third critical function test attempts to unwrap the HSM Device Integrity Key. The HDIK, post initialization, is wrapped with the HSM Device Integrity Key Encryption Key that is derived from the HSDK. If the HDIK fails to unwrap, or if the HDIK does not validate against the key contained in the HDIC, then the module is put into the factory reset state, in which all user data is cleared and the HSDK, HCK and HCC are erased.

The fourth critical function test involves unwrapping the HSM Communications Key and validating it against the key stored in the HSM Communications Certificate. The HCK is wrapped with the HSM Communications Key Encryption Key which is derived from the HSDK. If the HCK fails to unwrap, or if the HCK does not validate against the key contained in the HCC, then the module is put into the factory reset state, in which all user data is cleared and the HSDK, HCK, and HCC are all erased.

When the module is initialized, in which the module is brought out of the factory reset state, another critical function test is performed. Before the module is initialized (it is in the factory reset state), the HDIK is wrapped with the HSM Device Integrity Key Encryption Key which is derived from the HSM Administrator's password. If the HDIK cannot be unwrapped during device initialization, or if the HDIK does not validate against the key contained in the HSM Device Integrity Certificate, then the module is in

an un-recoverable error state and must be returned to the manufacturer.

10. EMI/EMC

The Envieta QFlex HSM conforms to the EMI/EMC requirements specified by part 47 Code of Federal Regulations, Part 15, Subpart B, Unintentional Radiators, Digital Devices, Class B.

11. Guidance and Secure Operation

11.1 Crypto Officer Guidance

To fully utilize the Envieta QFlex HSM, a user will need to use the provided Cryptoki Library that is PKCS#11 2.40 compliant, an administration application, and, optionally, smart cards.

11.1.1 Envieta PKCS#11 Library

The Envieta PKCS#11 Library is compliant with version 2.40 of the PKCS#11 standard. This library interfaces with the HCSA running on the HSM. Through this library, operators can access the approved cryptographic functions of the HSM.

11.1.2 Administration Program and Smart Cards

The Envieta HSM Administration Program directly communicates with the HASA. The administration program is used by the HSM administrator or by slot cryptographic officers to perform maintenance and administration of the HSM. It is through this interface that smart cards can be utilized. Using smart cards, users can login and import and extract keys to and from the HSM, using encrypted key splits.

11.1.3 Initializing the Module and Verifying Module Status

Upon receipt of the module, the administrator should ensure that both the outer cardboard packaging and inner static bags have intact tamper proof seals. Packaging should be inspected for any signs of damage, and the hardware serial numbers compared to the packing list. Afterwards, the administrator should install and initialize the device.

1. Install the device in an available PCIe Gen 3 slot with at least 8 lanes.
2. Power up the host machine and observe that LED#4 transitions to a blinking blue and white pattern.
3. Observe that LED #3 is cyan.
4. Using the administrative command line interface, initialize the device by providing a new administrative password and label for the device.
5. Observe that LED#3 is now solid green, indicating that the device is in the approved mode of operation.

The administrator can observe LED #3 and determine if the module is in the approved mode of operation if it is solid Green. If the module is in an error state, LED #3 will be solid red. The module will

also report its mode of operation through the Administrative interface in addition to the LED statuses.

11.2 Crypto Officer Guidance

Details on how to perform the security officer's role securely once the HSM is installed are contained in the QFlex Administrative User's Guide. Specifically, the following rules must be adhered to:

1. If using QFlex HSM Rev B, generate the Recoverable Tamper Key and perform a backup. If the backup is lost, this can result in permanent loss of the HSM contents.
2. Generate the HSM / Slot Backup Keys as appropriate and make a backup of the keys. Perform regular backups of these keys, as key material is added to slots.
3. Select a complex and secure password that is compliant with Section 5.2.1 above.
4. Assign users with the minimum required permissions to perform their intended functions.
5. To perform firmware updates, login as HSM administrator using the CLI. Load the firmware into the module using the "load_firmware_image" command which performs the cryptographic validation of the update image. Once validated, run the "perform_firmware_update" command. Only firmware versions that have been validated by the CMVP are allowed to be used.

11.3 Operator Guidance

The module design corresponds to the security rules below. The term *must* in this context specifically refers to a requirement for correct usage of the module in the Approved mode; all other statements indicate a security rule implemented by the module.

1. The module clears previous authentications on power cycle.
2. Power up self-tests do not require any operator action.
3. Data output is inhibited during key generation, self-tests, zeroization, and error states.
4. Status information does not contain CSPs or sensitive data that if misused could lead to a compromise of the module.
5. There are no restrictions on which keys or CSPs are zeroized by the zeroization service.
6. The module does not support a maintenance interface or role.
7. The module does not support manual key entry.
8. The module does not output intermediate key values.
9. The module requires split knowledge procedures using multiple authenticated operators prior to outputting plaintext CSPs.
10. The cryptographic officer must retain control of the module while zeroization is in process.
11. The permissions as described in Section 5 above must use the defaults when assigning a new operator to the module. While an operator with the "User Management" permission may assign customized permissions to a role, the access control functionality was tested using only the default permissions.
12. In addition to configuring specific permissions for each operator, it is possible to require multiple operators with permissions to be authenticated simultaneously in order to exercise certain permissions. An operator with the HSM Management permission can configure how many operators must be authenticated in order to exercise an operation. This is to ensure

security critical functionality cannot be accomplished with a single user if that level of control is desired by an end user.

Glossary

Term	Description
AES	Advanced Encryption Standard
ARM	Advanced RISC Machine
CAVP	Cryptographic Algorithm Validation Program
CKG	Cryptographic Key Generation
CMVP	Cryptographic Module Validation Program
CRNGT	Continuous Random Number Generator Test
CSP	Critical Security Parameter
CTR	Counter
DRBG	Deterministic Random Bit Generator
DSA	Digital Signature Algorithm
ECDSA	Elliptic Curve Digital Signature Algorithm
EMI/EMC	Electromagnetic Interference / Electromagnetic Compatibility
FIPS	Federal Information Processing Standards
FPGA	Field Programmable Gate Array
GCM	Galois/Counter Mode
HMAC	Hashed Message Authentication Code
HSM	Hardware Security Module
IG	Implementation Guidance
IV	Initialization Vector
KAS-SSC	Key Agreement Scheme-Shared Secret Computation
KAT	Known Answer Test
KBKDF	Key-Based Key Derivation Function
KDA	Key Derivation Algorithm
KDF	Key-Derivation Function
KTS	Key-Transport Scheme
LED	Light Emitting Diode
NIST	National Institute of Standards and Technology
NDRNG	Non-Deterministic Random Number Generator
OAEP	Optimal Asymmetric Encryption Padding
PBKDF	Password-Based Key Derivation Function
PCB	Printed Circuit Board
PCIe	Peripheral Component Interconnect Express
RSA	Rivest Shamir Adleman
SHA	Secure Hash Algorithm
SHS	Secure Hash Standard
UART	Universal Asynchronous Receiver-Transmitter
USB	Universal Serial Bus

Table 11 – Glossary of Terms