FIPS 140-2 Non-Proprietary Security Policy Pitney Bowes X4 Hardware Security Module (HSM)

KMS 6
Mailing Solutions Management Engineering
Version 02.00
10/01/18

Pitney Bowes, Inc
37 Executive Drive
Danbury, CT 06810

Contents

1. Module Overview

This document describes the Security Policy for the Pitney Bowes X4 Hardware Security Module (HSM) cryptographic module, created by Pitney Bowes, Inc. (PB).

Table 1 – Pitney Bowes X4 HSM components

Item	**Version**
Pitney Bowes X4 Hardware Security Module (HSM)	Part # 4W84001 Rev AAA
Hardware: MAX32590 Secure Microcontroller	Revision B4
Firmware components:	
Device Abstraction Layer (DAL)	01.01.0103
PRNG Library	01.01.0009
AES Library	01.01.0008
ECDSA Library	01.01.000A
DSA Library	01.01.000A
HMAC Library	01.01.0008
KAS Library	01.01.0008
DH Library	01.01.0008
RSA Library	01.01.000C
Hash Library	01.01.0008
Common Crypto Library	01.01.000B
Bootloader Interface Library	00.00.000F
PB Bootloader	00.00.0016
HSM Application	21.01.0021

The X4 Hardware Security Module provides cryptographic services to a host device. These include Authentication, Privacy, and Key Protection.

The X4 HSM is defined as a single chip cryptographic module.

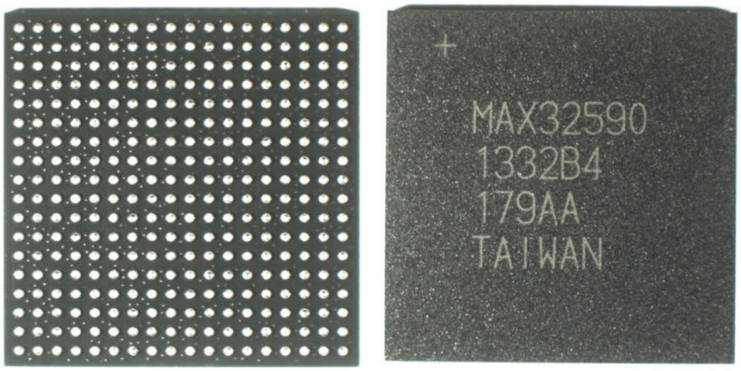

Figure 1 - MAX32590 Secure Processor (Back and Front)

The X4 HSM's cryptographic boundary is defined as the package that comprises the Maxim Integrated MAX32590 secure microcontroller. PB executable code is stored in external memory and copied to internal SRAM to be executed. On each power-up the firmware components listed in Table 1 are copied to internal SRAM and then authenticated via digital signatures:

Once the PB Bootloader has been loaded and authenticated, the PB Bootloader copies the combined Device Abstraction Layer (DAL) and Application to SRAM and authenticates it using an ECDSA P-256 with SHA-256 signature verification. The ECDSA P-256 SigVer function part of the PB Bootloader has been CAVP validated (Algorithm Cert. #529).

2. Security Level

The X4 HSM meets the overall requirements applicable to Level 3 security of FIPS 140-2.

Table 2 - Module Security Level Specification

Security Requirements Section	Level
Cryptographic Module Specification	3
Module Ports and Interfaces	3
Roles, Services and Authentication	3
Finite State Model	3
Physical Security	3 + EFP
Operational Environment	N/A
Cryptographic Key Management	3
EMI/EMC	3
Self-Tests	3
Design Assurance	3
Mitigation of Other Attacks	3

3. Modes of Operation

The X4 HSM uses FIPS approved algorithms contained in the DAL. The DAL supports the following FIPS Approved algorithms:

Table 3 – Approved Algorithms

CAVP Cert.	Algorithm	Standard	Mode/ Method	Key Lengths, Curves, or Moduli	Use
2826	AES	FIPS 197	ECB, CBC	128, 192, 256	Used to encrypt data output from the module and decrypt data input into the module.
2936[1]	AES	FIPS 197 SP 800-38F	KW	128, 192, 256	Key Wrapping/Unwrapping. Used to protect secret data. (The key establishment methodology provides 128, 192, or 256 bits of encryption strength)
Vendor affirmed	CKG	SP 800-133			Key generation from the unmodified output of the DRBG
334	CVL	SP 800-56A		P-256	ECC CDH primitive for key agreement.
1138	CVL	FIPS 186-4	SHA-256	P-256	ECDSA component for signature generation.

[1] AES Certificates #2826 and #2936 apply to the same AES implementation

CAVP Cert.	Algorithm	Standard	Mode/ Method	Key Lengths, Curves, or Moduli	Use
487	DRBG	SP 800-90A	HASH-based		Deterministic Random Bit Generator
871	DSA	FIPS 186-4	SHA-256[2]	2048	Generation of cryptographic key pairs, digital signature generation and verification
529	ECDSA	FIPS 186-4	SHA-256	P-256[3]	Generation of cryptographic key pairs, digital signature generation and verification for P-256
1769	HMAC	FIPS 198-1	HMAC-SHA-256[4]	256	Used to generate Message Authentication Codes (MACs). Truncated MACs (128 bits) are used for some applications, including user authentication.
49	KAS EC-DH	SP 800-56A	ECC	P-256	Key Agreement Protocol used to establish a session key (Ephemeral Unified Model C (2, 0, ECC CDH))
1539	RSA	FIPS 186-4	KeyGen X9.31 (2048) SigGen X9.31[5] (2048 w/SHA-256) SigVer X9.31 (2048 w/SHA-256) SigGen PKCSv1.5 (2048 w/SHA-256) SigVer PKCSv1.5 (2048 w/SHA-256) SigGen PSS (2048 w/SHA-256) SigVer PSS (2048 w/ SHA-256)		Used for key generation, digital signature generation and encryption (key encapsulation).
2369	SHS	FIPS 180-4	SHA-1, SHA-256		SHA-1 provides the hashing algorithm for key derivation. SHA-256 provides the hashing algorithm used as part of the digital signature process for RSA, DSA and ECDSA and in the generation of HMAC-SHA-256 message authentication codes.

[2] SHA-1 with L=1024 Signature Verification is included in the CAVS certificate, but not used in FIPS mode by any services

[3] SHA-1 with P-192 Signature Verification is included in the CAVS certificate, but not used in FIPS mode by any services

[4] HMAC-SHA-1 (key length 128) is included in the CAVS certificate, but not used in FIPS mode by any services

[5] The following are included in the CAVS certificate, but are not used in FIPS mode by any services: SigVer X9.31 (2048 w/SHA-1 and 1024 w/SHA-1, SHA-256); SigVer PKCSv1.5 (2048 w/ SHA-1 and 1024 w/SHA-1, SHA-256), SigVer PSS (2048 w/ SHA-1 and 1024 w/SHA-1, SHA-256)

The module supports the following non-Approved but Allowed security functions:

Table 4 – Non-Approved but Allowed Security Functions

Algorithm	**Caveat**	**Use**
NDRNG	The module generates cryptographic keys whose strengths are modified by available entropy. The NDRNG produces at least 211 bits of entropy for use in key generation.	Seeding of the DRBG

The module supports the following non-Approved security functions while operating in non-Approved mode:

Table 5 - Non-Approved Security Functions

Algorithm	**Use**
Diffie Hellman	Key Agreement Protocol used to establish a session key. Key agreement using 1024 bit keys (non-Approved mode of operation only)
DSA (non-compliant)	This algorithm is used to generate key pairs and generate signatures for L=1024, N=160 & SHA-1 (non-Approved mode of operation only).
ECDSA (non-compliant)	This algorithm is used to generate key pairs, digital signatures for P-160 (SHA-1) and P-192 (SHA-1) curves (non-Approved mode of operation only).
RSA (non-compliant)	This algorithm is used to generate keys and digitally sign using schemes PKCS 1.5, X9.31 and PSS, PKCS 1 version 2.1 for 1024 bit modulus using SHA-1 (non-Approved mode of operation only).
Triple-DES (non-compliant)	Encryption using 2-Key Triple-DES (non-Approved mode of operation only)
Triple-DES MAC	128-bit and 192-bit Triple DES MAC Generation per FIPS 113 (non-approved mode of operation only)

3.1 FIPS Mode Indicator

The module supports both an Approved mode and a non-Approved mode of operation. The module provides an explicit mode of operation indicator that reflects the mode of operation. The module's "Get Status" service returns the FIPS mode status flag. The FIPS mode flag is set to zero for FIPS Approved mode or one for non-Approved mode of operation.

4. Ports and Interfaces

The MAX32590 is supplied in a 324-pin BGA package where all power input, data input, data output, control input, and status output interfaces are supported.

		Ball Grid Array Pin Horizontal from "x"																	
		1	***2***	***3***	***4***	***5***	***6***	***7***	***8***	***9***	***10***	***11***	***12***	***13***	***14***	***15***	***16***	***17***	***18***
Ball Grid Array Pin	***A***	-	-	-	-	-	-	O	-	-	-	-	-	-	-	-	-	-	-
	B	-	-	-	-	-	-	I	-	-	-	-	-	-	-	-	-	-	-
	C	-	-	P	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	D	-	-	P	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	E	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

	F	-	-	-	-	-	P	P	P	P	P	P	P	IO	IO	-	-	-	-
	G	-	-	-	-	S	P	-	-	-	-	-	P	C	-	-	-	-	-
	H	-	-	-	-	P	P	-	-	-	-	-	P	S	-	-	-	-	-
	J	-	-	-	-	C	P	-	-	-	-	-	P	C	-	-	-	-	-
	K	-	-	-	-	C	P	-	-	-	-	-	P	C	-	-	-	-	-
	L	-	-	-	-	-	P	-	-	-	-	-	P	-	-	-	-	-	-
	M	-	-	-	S	-	P	-	-	-	-	-	P	-	-	-	-	-	-
	N	-	-	-	-	-	P	P	P	P	P	P	P	-	S	-	-	S	S
	P	-	-	-	O	-	O	O	O	O	O	O	-	O	O	O	O	O	O
	R	-	-	-	-	-	-	-	-	-	IO	IO	IO	IO	O	O	O	O	O
	T	-	-	-	-	-	-	-	-	-	IO	IO	IO	IO	-	O	O	O	O
	U	-	-	-	-	-	-	-	-	-	IO	IO	IO	IO	-	-	O	O	O
	V	-	-	-	-	-	-	-	-	-	IO	IO	IO	IO	-	-	O	O	O

I = Data In O = Data Out S = Status Out C = Control In P = Power - = Disabled

Figure 2 – Interface Mapping

5. Identification and Authentication Policy

The module supports three authenticated roles, that are either a Crypto-Officer (CO) or User. All services described in Section 6 below (except the service available to Crypto-Officer (Administrator)) are available to both the CO (Operator) and Trusted User (TU). Additionally, the module has an Unauthenticated User role.

Table 6 - Roles and Authentication Type

Role	***Authentication Method***	***Authentication Type***
Crypto-Officer (Administrator)	*Digital Signature (ECDSA P-256)*	*Identity-based*
Crypto-Officer (Operator)	*Uniquely Assigned ID in conjunction with HMAC-SHA-256 Secret Key*	*Identity-based*
Trusted User (TU)	*Uniquely Assigned ID in conjunction with HMAC-SHA-256 Secret Key*	*Identity-based*
Unauthenticated User	*None*	*None*

Table 7 - Authentication Strength

Authentication Mechanism	***Strength Mechanism***
Unique signature	*Authentication of the Crypto-Officer (Administrator) is based on the verification of a unique ECDSA (P-256) signature. This provides 128 bits of security.*
ID and Secret Cryptographic Key Combination	*Authentication of the Crypto-Officer (Operator) and the Trusted User is based on a unique (256-bit) HMAC-SHA-256 secret cryptographic key (DAK), with a MAC truncated to 128 bits. This provides 128 bits of security.* *The probability of a random access or false acceptance occurring is 1 in 2^128 for ECDSA or the truncated HMAC-SHA-256 MAC, which is less than 1 in 1,000,000.* *The module can execute at most 17.85 ECDSA verifications per second or 3,000 HMAC-SHA-256 authentication attempts per second. Therefore, the probability of a successful random attempt in a one minute period is 1 in 3.2 x 10^35 for ECDSA and 1 in 1.9 x 10^33 for HMAC-SHA-256. Both of these values are far less than 1 in 100,000.*

6. Access Control Policy

Each service described below is available to both the Crypto-Officer (Operator) and Trusted User roles (except the service available to Crypto-Officer (Administrator)). Unauthenticated services are also available to the Unauthenticated User role.

Crypto Officer (Administrator) Services:

- **Firmware Update** is described in section 7.1 Firmware Update

Crypto Officer (Operator) and Trusted User (TU) Services:

- **Generate**: The Crypto Officer or Trusted User sends this block to instruct the X4 HSM to generate a Public/Private key pair or a Secret key. The message specifies the cryptographic algorithm and the parameters for use in the generation of the key(s).
- **Load Key**: The Crypto Officer or Trusted User sends this command to instruct the X4 HSM to load an encrypted key record for later use. The command specifies the storage type:
  - Volatile: Store in RAM, can be replaced if space is needed.
  - Sticky: Store in RAM, can NOT be replaced until it is deleted by the host.
  - Static: Store in NVM
- **Split Key**: The Crypto Officer or Trusted User sends this command to instruct the X4 HSM to divide a key into 2 or more parts.
- **Join Key**: The Crypto Officer or Trusted User sends this command to instruct the X4 HSM to assemble a key that has been previously split.
- **Export Key**: The Crypto Officer or Trusted User sends this command to have a key securely exported for storage / use in another location.
- **Encrypt**: The Crypto Officer or Trusted User sends this command to encrypt data.
- **Decrypt**: The Crypto Officer or Trusted User sends this command to decrypt data.
- **Load Parameters**: The Crypto Officer or Trusted User sends this message to load a set of parameters into the HSM.
- **Derive Key**: The Crypto Officer or Trusted User sends this message to generate a key to be used by the host and the HSM to exchange secure information.
- **Decrypt Encrypt**: The Crypto Officer or Trusted User sends this message to allow encrypted data to be decrypted and re encrypted with another key.
- **Decrypt Compare:** The Crypto Officer or Trusted User sends this message to allow encrypted data to be decrypted and compared.
- **Sign**: The Crypto Officer or Trusted User sends this message to apply a cryptographic signature to a set of data.

- **Verify**: The Crypto Officer or Trusted User sends this message to verify a cryptographic signature on a set of data.
- **Get Counters:** Instructs the HSM to output a copy of the current values of the internal counters.
- **Set Counter**: The Crypto Officer or Trusted User sends this message to set an internal counter to a specific value.
- **Update Counters:** Instructs the HSM to update specific internal counters
- **Generate PSD RSA Record:** The Crypto Officer or Trusted User sends this block to instruct the X4 HSM to generate an RSA Public/Private key pair record for a Postal Security Device.
- **Generate RSA Primes:** The Crypto Officer or Trusted User sends this block to instruct the X4 HSM to generate a pair of prime numbers used for RSA key generation.
- **Delete Key**: The Host sends this message to remove a key from the HSM.

Unauthenticated Services:

Miscellaneous functions that do not require X4 HSM authentication of the entity; These services are available to all roles.

- **Get Random**: The Host sends this message to get a pseudo-random number from the HSM.
- **Get Key List**: Instructs the HSM to return a list of all active keys stored in the HSM.
- **Get Parameters:** Instructs the HSM to retrieve parameter values from the HSM. The Host can request individual parameter IDs or all of the stored parameters in the HSM.
- **Perform Diagnostic Test:** Instructs the HSM to request that the X4 HSM perform one or more diagnostic tests.
- **Read Log:** Instructs the HSM to get Log Data. The number of available entries, the size of each entry, and the data contained in each entry will depend on the log that is being requested.
- **Get Status:** Instructs the HSM to request HSM status information.
- **Get HW Status:** Instructs the HSM to request the hardware specific status data of the HSM.
- **Reboot:** Instructs the HSM to reboot the application.
- **Get Versions:** Instructs the HSM to get the versions of the components in the HSM
- **Reinit:** Instructs the X4 HSM to erase all NVM data except for HW Mfg Data and ‘persistent’ data (total device cycles, reinit count) and then invalidates the HSM DAL. This command zeroizes the Unique HSM Key Encryption Key and Unique HSM Key Authentication Key, which result in the loss of all other Private and Secret Keys. Used to ‘clean’ the HSM so it can be re-configured.

7. Firmware Update Access Control Policy

The HSM supports a secure firmware update process. The Software Download Utility (SDU) within the Device Abstraction Layer is responsible for firmware updates. The SDU is also used at power-up (described in Section 1.0) to load the firmware components necessary to operate the module.

7.1 Combined App/DAL Download

The Bootloader loads combined App/DAL in chunks. Each chunk is verified with signed hash record containing an ECDSA P-256 (Cert. #529) signature. When SDU download is complete a hash is calculated and verified on the entire App/DAL firmware image.

The Software Download Utility supports the following messages:

Crypto-Officer (Administrator)

- Setup Download Data: The Host sends this signed record to make the Software Download Utility aware of the parameters of the software (application) to be downloaded. This message is signed by the SWAK (Software Authentication Key). Receipt of this message triggers a transition to the state required to load chunk information.
- Setup Download Chunk: The Host sends this signed record to make the Software Download Utility aware of the parameters of the software (application) chunk to be sent in the following message. Receipt of this message triggers a transition to the state required to load the chunk. The Setup Download Chunk message is only valid if the DAL has received a valid Setup Download Data message.
- Download Chunk: This message contains the data referenced in the Setup Download Chunk message.

Utility Functions

The following utility functions are unauthenticated and intended to aid the host application in managing the software update process.

- Reboot: This function is used to invoke a reboot. It returns a 'Reboot' response message, waits until the transmit channel is idle then resets the ASIC.
- Initialize: This function writes 0's to the DAL Software Validity Flag, sends a response message, waits for until the transmit channel is idle and then resets the ASIC. The HSM transitions to the ROM Firmware State after completion of the reboot.
- Get Status: The Host device sends this message to the HSM to request the current status information.

8. Definition of Critical Security Parameters (CSPs)

There are five CSPs that are necessary for the HSM to function as a FIPS device. Other keys may be loaded or generated by command from the user. These keys could include DSA, ECDSA, AES, RSA, HMAC, EC-DH private/secret or public keys to meet the needs of the Crypto-Officer or Trusted User while using the HSM. All private and secret keys that are stored as ciphertext are encrypted by the KEK (256-bit AES, Cert. #2936). All transported secret and private keys are encrypted by AES (Cert. #2936) per SP 800-38F. The KAK is used to authenticate keys and the DAK is used to authenticate identities.

The first five entries in the following table describe the five necessary CSPs contained in the module:

Table 8 - CSPs

Key	Key Name	Description / Usage	Generation / Agreement	Storage	Entry / Output	Destruction
KEK	Unique HSM Key Encryption Key	AES256Key Encryption Key	Internally by FIPS approved DRBG	Clear text	Entry: N/A Output: N/A	Zeroized on Tamper or Reinitialize or removal of all power
KAK	Unique HSM Key Authentication Key	HMAC256Key Authentication Key	Internally by FIPS approved DRBG	Ciphertext	Entry: N/A Output: N/A	Zeroized on Tamper or Reinitialize or removal of all power
DAK	DAL Authentication Key	HMAC256 Key	Entered in factory environment	Ciphertext	Entry: N/A Output: Encrypted	Encrypting key zeroized on Tamper or Reinitialize or removal of all power
DPK	DAL Privacy Key	AES256 Key	Entered in factory environment	Ciphertext	Entry: N/A Output: Encrypted	Encrypting key zeroized on Tamper or Reinitialize or removal of all power

Key	Key Name	Description / Usage	Generation / Agreement	Storage	Entry / Output	Destruction
DRBG V	DRBG Seed	DRBG seed	Internally generated by NDRNG	Ciphertext	Entry: N/A Output: N/A	Encrypting key zeroized on Tamper or Reinitialize or removal of all power
DRBG WS	DRBG Working State	Internal state of DRBG	Internally generated by DRBG	Ciphertext in BRAM	Entry: N/A Output: N/A	Encrypting key zeroized on Tamper or Reinitialize or removal of all power
AK	Authentication Key	HMAC256	Internally by FIPS approved DRBG or Loaded as needed	Ciphertext	Entry: Encrypted Output: Encrypted	Encrypting key zeroized on Tamper or Reinitialize or removal of all power
		DSA2048	Internally by FIPS approved DRBG or Loaded as needed	Ciphertext	Entry: Encrypted Output: Encrypted	
		ECDSA256	Internally by FIPS approved DRBG or Loaded as needed	Ciphertext	Entry: Encrypted Output: Encrypted	
		RSA2048	Internally by FIPS approved DRBG or Loaded as needed	Ciphertext	Entry: Encrypted Output: Encrypted	
PK	Privacy Key	AES128	Internally by FIPS approved DRBG or Loaded as needed	Ciphertext	Entry: Encrypted Output: Encrypted	

Key	Key Name	Description / Usage	Generation / Agreement	Storage	Entry / Output	Destruction
		AES192	Internally by FIPS approved DRBG or Loaded as needed	Ciphertext	Entry: Encrypted Output: Encrypted	Encrypting key zeroized on Tamper or Reinitialize or removal of all power
		AES256	Internally by FIPS approved DRBG or Loaded as needed	Ciphertext	Entry: Encrypted Output: Encrypted	
DPAG Private	DPAG Private Key	RSA2048 Private Key	Internally by FIPS approved DRBG or Loaded as needed	Ciphertext	Entry: Encrypted Output: Encrypted	Encrypting key zeroized on Tamper or Reinitialize or removal of all power
EC-DH	EC-DH	ECC-CDH Private Key	Generated	Plaintext	Entry: N/A Output: N/A	Immediately after Session established
SS	Shared Secret	Used to derive Session Key (SK)	ECC-CDH Key Agreement	Plaintext	Entry: N/A Output: N/A	Immediately after Session established
SK	Session Key	AES128	Derived from Shared Secret	Ciphertext	Entry: N/A derived internally Output: Encrypted or Never output	Encrypting key zeroized on Tamper or Reinitialize or removal of all power
		AES192	Derived from Shared Secret	Ciphertext	Entry: N/A derived internally Output: Encrypted or Never output	
		AES256	Derived from Shared Secret	Ciphertext	Entry: N/A derived internally Output: Encrypted or Never output	

The following table describes the public keys contained in the module:

Table 9 – Public Keys

Key	Key Name	Description / Usage	Generation / Agreement	Storage	Entry / Output
SWAK	HSM SW Download Key	ECDSA P-256 used to validate firmware	Externally	Plaintext	Entry: Hard coded in SDU Output: N/A
EC-DH Public	ECC-CDH Public	ECC-CDH Public Key	Generated	Plaintext	Entry: N/A Output: Plaintext
DPAG Public	DPAG Public Key	RSA2048 Public Key	Internally by FIPS approved DRBG or Loaded as needed	Plaintext	Entry: Plaintext Output: Plaintext
AK Public	Public Authentication Key	DSA2048	Internally by FIPS approved DRBG or Loaded as needed	Plaintext	Entry: Plaintext Output: Plaintext
		ECDSA256	Internally by FIPS approved DRBG or Loaded as needed	Plaintext	Entry: Plaintext Output: Plaintext
		RSA2048	Internally by FIPS approved DRBG or Loaded as needed	Plaintext	Entry: Plaintext Output: Plaintext

The following table describes the modes of access for each key to each role supported by the module. The CO role refers to the Crypto-Officer (Operator). The Crypto-Officer (Administrator) has access to the SWAK (a public key) through the Authenticated Service Firmware Update. The modes of access are defined as:

Table 10 – CSP Modes of Access

Roles		**Services**	**CSP Modes of Access**
CO	***TU***	***Authenticated Services***	
X	X	Generate	Generates AK and PK
X	X	Load Key	Loads AK and PK
X	X	Split Key	Splits AK and PK
X	X	Join Key	Joins AK and PK
X	X	Export Key	Exports AK and PK
X	X	Encrypt	Uses PK
X	X	Decrypt	Uses PK
X	X	Load Parameters	N/A
X	X	Derive Key	Uses AK and PK
X	X	Decrypt Encrypt	Uses PK
X	X	Decrypt Compare	Uses PK
X	X	Sign	Uses AK
X	X	Verify	Uses AK
X	X	Get Counters	Uses AK
X	X	Set Counter	N/A
X	X	Update Counters	N/A
X	X	Generate PSD RSA Record	Generates AK
X	X	Generate RSA Primes	Generates DPAG Private
X	X	Delete Key	Used to remove AK and PK
CO	***TU***	***Unauthenticated Services***	
X	X	Get Random	N/A
X	X	Get Key List	N/A
X	X	Get Parameters	N/A

X	X	Perform Diagnostic Test	N/A
X	X	Read Log	N/A
X	X	Get Status	N/A
X	X	Get HW Status	N/A
X	X	Reboot	N/A
X	X	Get Versions	N/A
X	X	Reinit	Zeroizes Secret and Private key data

9. Operational Environment

The FIPS 140-2 Area 6 Operational Environment requirements for the module are not applicable because the device does not contain a modifiable operational environment.

10. Security Rules

This section documents the security rules enforced by the module to implement the security requirements of this FIPS 140-2 Level 3 module.

- The module shall not process more than one request at a time (i.e., single threaded). While processing a transaction, prior to returning a response, the module will ignore all other inputs to the module. No output is performed until the transaction is completed, and the only output is the transaction response.
- The module shall validate identities using digital signatures and MACs.
- All keys generated in the module shall have at least 112 bits of strength for FIPS Approved mode of operation.
- All methods of key generation shall be at least as strong as the key being generated.
- The module shall not provide a bypass state where plaintext information is just passed through the module.
- The module shall not support a maintenance mode.
- The module shall not output any secret or private key in plaintext form.
- The module shall not accept any secret or private key in plaintext form outside of manufacturing.
- There shall be no manual entry of keys into the system.
- There shall be no entry or output of split keys from the system except when the HSM is configured as a Key Root Authority (KRA).
- Keys shall be either generated via an Approved method or entered into the system through FIPS Approved processes.

- Once a module has been zeroized, it must be returned to the factory for software loading and parameterizing prior to being usable by a customer.

10.1 Self-Tests

10.1.1 Conditional

The module shall support conditional tests, which can also be run as requested by the user, including:

- Pairwise consistency tests:
  - DSA 2048 pairwise consistency test for key pair generation
  - ECDSA P-256 pairwise consistency test for key pair generation
  - RSA 2048 pairwise consistency test for key pair generation
- Random number generator health test:
  - NDRNG Continuous NDRNG test
- Software/Firmware Load Test:
  - ECDSA P-256 Signature Verification (Cert. #529)
- ECDSA P-256 Public Key Validation as part of SP 800-56A Key Agreement Protocol

10.1.2 Power-up

The module shall support power up self-tests, which can also be run as requested by the user, including:

- Firmware Integrity Tests:
  - Digital Signature Verification – ECDSA P-256
- Bootloader Power On Self-Tests (POST):
  - ECDSA P-256 Signature Verification Known Answer Test
  - SHA-256 Known Answer Test
- Critical functions tests:
  - RTC Test
  - Bootloader Test
  - BRAM Pattern Test
- Cryptographic Algorithm Known Answer Tests: (DAL POST)
  - AES 256 Encrypt and Decrypt Known Answer Tests
  - DRBG SP800-90A Known Answer Tests (Instantiate, Generate)
  - DSA 2048 Signature Generation and Verification Known Answer Tests

- ECDSA P-256 Signature Generation and Verification Known Answer Tests
- HMAC SHA-256 Known Answer Test
- KAS SP800-56A (C(2e, 0s, ECC CDH)) Known Answer Test
- RSA 2048 Signature Generation and Verification Known Answer Tests
- SHA-1 Known Answer Test
- SHA-256 Known Answer Test

Self-tests may be initiated by the following means:

- Perform Diagnostic Test service
- Physically re-cycling the module's power

The status of self-tests shall be available via the Get Status service.

11. Physical Security Policy

The MAX32590 is a single chip cryptographic module which protects key material from unauthorized disclosure. The security features in the module include real time environmental monitoring (temperature, battery, voltage) and tamper detection. Triggering the environmental monitors or damaging the tamper shield results in a destructive result, which halts the processor and automatically zeroizes the internal encrypting key. The module protects Critical Security Parameters (CSPs). The operator should periodically inspect the module for evidence of tampering.

12. Mitigation of Other Attacks Policy

The module has been designed to mitigate specific attacks outside the scope of FIPS 140-2, Level 3. It incorporates environmental failure protection mechanisms inherent to that of a Level 4 module. The module is designed to defend against out of bound voltage and temperature extremes and against physical probing, as described in the Physical Security Policy.

13. References

The following documents are referenced by this document, are related to it, or provide background material related to it:

- Digital Signature Standard (DSA) – FIPS PUB 186-4, July, 2013
- Advanced Encryption Standard (AES) FIPS PUB 197, November 26, 2001
- Recommendation for Block Cipher Modes of Operation, Methods and Techniques, Special Publication 800-38A, December 2001.

- Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher, Special Publication 800-67, Jan 2012.
- The Keyed-Hash Message Authentication Code (HMAC), Federal Information Processing Standards Publication 198-1, July 2008
- Recommendation for Random Number Generation Using Deterministic Random Bit Generators, Special Publication 800-90A, January 2012.
- AES Key Wrap – NIST Special Publication 800-38F – December 21, 2012
- Secure Hash Standard – FIPS PUB 180-4, March 2012
- NIST SP 800-56A, Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography – March 2007
- Security Requirements for Cryptographic Modules – FIPS PUB 140-2, Change Notices December 3, 2002

14. Acronyms

AES	Advanced Encryption Standard
ASIC	Application-Specific Integrated Circuit
BRAM	Battery Backed RAM
CAVP	Cryptographic Algorithm Validation Program
CBC	Cipher Block Chaining
CO	Crypto Officer
CSP	Critical Security Parameters
CVL	Component Validation List
DAL	Device Abstraction Layer
DES	Data Encryption Algorithm
DH	Diffie-Hellman
DRBG	Deterministic Random Bit Generator
DSA	Digital Signature Algorithm
ECB	Electronic Code Book
ECC CDH	Elliptic Curve Cryptography Cofactor Diffie-Hellman
EC-DH	Elliptic Curve Diffie-Hellman
ECDSA	Elliptic Curve Digital Signature Algorithm
EFP	Environmental Failure Protection
EMC	Electromagnetic Compatibility
EMI	Electromagnetic Interference
FIPS	Federal Information Processing Standards
HMAC	Hashed Message Authentication Code
HSM	Hardware Security Module
KAS	Key Agreement Scheme
KRA	Key Root Authority
NVM	Non-Volatile Memory
PB	Pitney Bowes
POST	Power-on Self-test
PSS	Probabilistic Signature Scheme
RAM	Random Access Memory
ROM	Read-Only Memory
RSA	Rivest Shamir Adleman
RTC	Real Time Clock
SDU	Software Download Utility
SHA	Secure Hash Algorithm
SRAM	Static RAM
TRNG	True Random Number Generator
TU	Trusted User

Revision History

Version	Date	Revision Description
1.00	12/21/2016	Original Document
2.00	10/01/2018	Revised for clarity