Pitney Bowes, Inc.

X4i Postal Security Device (PSD)

FIPS 140-2 Non-Proprietary Security Policy

Horizon CSD-IMI

Small and Medium Business Solutions Group

Version 2.2

TABLE OF CONTENTS

1. Cryptographic Module Specification 3

1.1 Overview 3

1.2 Security Level 4

1.3 Modes of Operation 4

2. Module Ports and Interfaces 4

3. Roles, Services, and Authentication 5

3.1 Roles 5

3.1.1 Initialization 6

3.2 Services 6

3.2.1 Service Access to Security Functions and CSPs 10

3.3 Non-Approved Mode Roles and Services 13

3.4 Security Rules 13

4. Physical Security 14

5. Mitigation of Other Attacks 14

6. Operational Environment 14

7. Cryptographic Key Management 15

7.1 FIPS Approved Algorithms 15

7.2 FIPS Allowed Algorithms 16

7.3 FIPS Non-Approved Algorithms 17

7.4 CSPs and Keys 17

7.4.1 Critical Security Parameters 17

7.4.2 Public Security Parameters Keys 19

7.4.3 Zeroization 20

8. Self-Tests 20

8.1 Power on Self-Tests 20

8.2 Conditional Tests 21

Appendix A: References 22

Appendix B: Abbreviations and Definitions 23

TABLE OF TABLES

Table 1 – X4i Postal Security Device (PSD) Component Versions 3
Table 2 – Module Security Level 4
Table 3 – Roles and Authentication 5
Table 4 – Strength of Authentication 5
Table 5 – Services Available in FIPS Approved Mode 11
Table 6 – FIPS Approved Algorithms 15
Table 7 – FIPS Allowed Algorithms 16
Table 8 – FIPS Non-Approved Algorithms 17
Table 9 – Secret Keys, Private Keys, Cryptographic Key Components, and Other CSPs 18
Table 10 – Public Security Parameters 19
Table 11 – References 22
Table 12 – Abbreviations and Definitions 23

1. CRYPTOGRAPHIC MODULE SPECIFICATION

1.1 Overview

This document describes the Security Policy for the X4i Postal Security Device (PSD) (the X4i PSD). The X4i PSD is a single-chip cryptographic module designed by Pitney Bowes, Inc. (PB) to conform with FIPS 140-2 Level 3 + EFP requirements.

Table 1 – X4i Postal Security Device (PSD) Component Versions

Item	Version	
Hardware Components:		
MAX32590 Secure Microcontroller	Revision B4	
Firmware Components:		
PB Bootloader	00.00.0016	
PSD Application Device Abstraction Layer (DAL)	21.06.0013 01.02.0018	21.07.000A 01.02.0024

The PSD Application and DAL are compiled into a single firmware and integrity tested together. This single firmware is referred to as the PSD Application hereafter.

The X4i PSD provides cryptographic services to a host device (i.e. Digital Postage Meter), to support postage evidence in the form of an indicium. A PSD provides protection that includes ensuring the secrecy of critical security parameters (CSPs) such as cryptographic keys, and providing data integrity protection for funds relevant data items (FRDIs[1]) such as accounting data. CSPs and FRDIs reside inside the strong physical protections of the PSD.

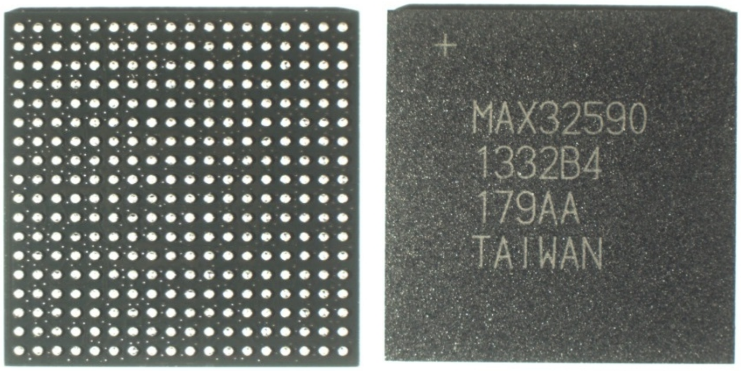

Figure 1 – MAX 32590 (Back and Front)

The X4i PSD's cryptographic boundary is defined as the IC package that comprises the Maxim Integrated MAX32590 DeepCover Secure Microcontroller (refer to Figure 1). PB executable code is stored in external memory and copied to internal SRAM to be executed. On each power up, the

[1] FRDIs are not applicable to FIPS 140-2 and are not CSPs. The FRDI's authenticity and integrity are critical for postal functionality and they should never be zeroized.

firmware components listed in Table 1 are copied to internal SRAM and then authenticated via digital signatures.

The PB Bootloader is authenticated by verification of the "CRK" key using RSA 2048 with SHA-256 (Cert. #C477). Once the PB Bootloader has been loaded and authenticated, the PB Bootloader copies PSD Application (i.e. the combined Device Abstraction Layer (DAL) and PSD Application) to SRAM and authenticates it by verification of the "SWAK" Key using ECDSA P-256 with SHA-256 (Cert. #C476).

1.2 Security Level

The module meets the overall requirements of FIPS 140-2 Security Level 3 +EFP

Table 2 – Module Security Level

FIPS Area	FIPS Security Requirement	Level
1	Cryptographic Module Specification	3
2	Module Ports and Interfaces	3
3	Roles, Services, and Authentication	3
4	Finite State Model	3
5	Physical Security	3 +EFP
6	Operational Environment	N/A
7	Cryptographic Key Management	3
8	EMI/EMC	3
9	Self-Tests	3
10	Design Assurance	3
11	Mitigation of Other Attacks	3

1.3 Modes of Operation

The module supports both an Approved mode and a non-Approved mode of operation. The module provides an explicit mode of operation indicator: the FIPS mode status flag is returned in every response from the module. The FIPS mode flag is set to zero for an Approved mode of operation or to one for non-Approved mode of operation.

The module's mode of operation can only be configured within manufacturing. Once configured, the module does not have the ability to change modes.

2. MODULE 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 Vertical from “x”	***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 – X4i PSD Interface Mapping

3. ROLES, SERVICES, AND AUTHENTICATION

3.1 Roles

The module supports three authenticated roles that are either categorized as Crypto-Officer (CO), or User roles. Additionally, the module has a single unauthenticated role. The CO role is implicitly selected and authenticated via digital signatures. The User role is explicitly selected by the Login Request service. The User (C+) role is implicitly selected by ID and possession of the HMAC key used for authentication.

Table 3 – Roles and Authentication

Role	Authentication Method	Authentication Type
Crypto-Officer	Digital Signature (ECDSA P-256, authenticated with Vendor, Download or Certificate Keys)	Identity-based
User	Uniquely Assigned ID in conjunction with 128-bit password	Identity-based
User (C+)	Unique ID in conjunction with HMAC-SHA-256 Session Authentication Key (truncated to 64 bits)	Identity-based
Unauthenticated	None	None

Table 4 – Strength of Authentication

Authentication Mechanism	Probability of False Acceptance (Single Attempt)	Probability of False Acceptance (One Minute)
Digital Signature	The probability of a random access or false acceptance occurring is 1 in 2^128 for ECDSA P-256, which is less than 1 in 1,000,000.	The module can execute at most 17.85 ECDSA verifications 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, which is far less than 1 in 100,000.
User: ID and Password Combination	The probability of a random access or false acceptance occurring is 1 in 2^128 for a given ID, which is less than 1 in 1,000,000.	The module can execute at most 40password authentication attempts per minute. Therefore, the probability of a successful random attempt in a one-minute period is 1 in 8.5 x 10^36, which is far less than 1 in 100,000.
User (C+): ID and HMAC	A 256-bit HMAC key (Session Authentication Key) truncated to 64 bits is used for authentication. The probability of a random access or false acceptance occurring is 1 in 2^64 for a given ID, which is less than 1 in 1,000,000.	The module can execute at most 3,000 HMAC authentication attempts per second. Therefore, the probability of a successful random attempt in a one-minute period is 1 in 1.0 x 10^14, which is far less than 1 in 100,000

3.1.1 Initialization

During manufacturing, the PB Bootloader (verified by CRK key) and PSD application (verified by SWAK key) are loaded at secure vendor facilities. The system is initialized, and the mode of operation is locked. The DRBG seed, KEK (Key Encryption Key) and KAK (Key Authentication Key) are generated. Signed public keys, including keys for authenticating the CO authentication keys, are loaded and verified. The module is configured. The Operation key pair (ECDSA) and Debit key pairs (ECDSA or DSA) are generated, and the public keys are signed and exported.

At the customer site, the Crypto-Officer is implicitly authenticated for all relevant services. The module loads the CRL and the User (either User role or User (C+) role) logs in and completes configuration. The User logs in with the assigned unique ID and either a password (pre-loaded on the module and specified through a separate secure channel) or a HMAC secret key (established through key agreement with the module by the CO).

3.2 Services

Crypto-Officer:

The Crypto-Officer is responsible for the high-level key management within the PSD. Its primary functions are to load keys into the PSD and to authorize the generation and use of the Debit and Operation Keys. The Crypto-Officer also manages non-key data used to set internal parameters and settings in the PSD. The PB Infrastructure or Manufacturing systems are the only entities who act as the PSD Administrator.

The services allocated to this role are as follows:

- **Generate PSD Key:** Instructs the PSD to generate its Unique (ECDSA P-256) *Operation Key* pair or the Unique (DSA 2048 or ECDSA P-256) *Debit Key* pair. The message contains a Signed Parameter Record with the parameters for use in the generation of the private and public key values. The algorithm used is determined by the Key Descriptor in the Signed Parameter Record and is based on postal requirements.

- **Generate Session Key:** Instructs the PSD to generate an AES 256-bit or a HMAC 256-bit session key via SP 800-56A conformant key agreement and SP 800-56C conformant key derivation that will be used during PB infrastructure communication sessions or to authenticate host communication sessions.
- **Load Certificate Key:** Instructs the PSD to load the (ECDSA P-256) *Certificate Key*. The PSD verifies the certificate with the *Vendor Key*.
- **Load CRL:** Loads the Certificate Revocation List and the CRL version. The PSD validates the signature on the CRL using the *Download Key*. If the CRL signature is valid and the version of the CRL is greater than or equal to any previously loaded CRL, the PSD stores the CRL in internal memory and stores the CRL version in Flash memory for future comparison. If validation fails, the PSD is disabled and an error is returned.

  Once the PSD is out of Manufacturing state, it will require that a CRL be loaded. Prior to loading a CRL, all functions requiring cryptographic operations other than Load CRL will be blocked. Any public key identified by the CRL will be blocked from use in the PSD.
- **Load Download Key:** Instructs the PSD to load the (ECDSA P-256) *Download Key* Certificate. The PSD verifies the certificate with the *Certificate Key*.
- **Load Encrypted Key:** The Crypto Officer instructs the PSD to load a signed key record containing an encrypted symmetric or private key.
- **Load Key Acknowledgement***:* Acknowledge that the generated PSD Key has been successfully registered and that the PSD can activate that key. The PSD verifies the Key Acknowledgement Block with the *Certificate Key*. If valid, the PSD activates the generated PSD key, allowing it to be used by the PSD.
- **Load Parameters:** Loads either functional or data parameters to the PSD. The parameter blocks are signed by the *Certificate Key*. If the PSD is in the *Operational* lifecycle state, the first parameter in the parameter block must be the challenge value from the most recent "Get Challenge" command to the PSD.

  Supported functional parameters are:

  - **Transition to Operational State:** Causes the PSD to transition to the PSD *Operational* lifecycle state.
  - **Transition to Base State:** Transitions the PSD from its *Manufacturing* lifecycle state to *Base* lifecycle state.
  - **Disable PSD:** Places the PSD in the *Disabled* lifecycle state. In the *Disabled* lifecycle state, further financial functions are prohibited.
  - **Enable PSD:** Transition the PSD from *Disabled* lifecycle state to *Operational* lifecycle state.
  - **Reinitialize PSD:** Causes PSD to zeroize all plaintext cryptographic keys and CSPs, and then invalidates the PSD Application. This command zeroizes the Unique PSD *Key Encryption Key* (KEK) which results in the loss of all Private and Secret Keys. The module must be returned to manufacturing after this point.

  - **Start Software Update:** Invalidates the current loaded PSD Application and jumps to the *Software Update Utility* entry point to allow start of software download with a new PSD Application. Only PB digitally signed software (verified by SWAK) can be authenticated and loaded. Only the CMVP-validated version of software can be loaded.
  - **Transaction Start:** Triggers event to have the PSD prepare for a multi-message transaction that must be completed successfully as a unit (atomic transaction). This means that if any one of the messages within the transaction fails, all messages must be rolled back.

    Not all messages sent after start of a transaction are processed to allow commit/rollback. The messages that are handled in the transaction are PVD (one occurrence), Load Parameters (only data parameters), Load Encrypted Key, and Generate PSD Key.

    - Transaction Commit: Triggers event to 'commit' the updates made by PVD, Load Parameters, Load Encrypted Key, and / or Generate PSD Key made after the Transaction Start event was processed.
    - Transaction Rollback: Triggers event to rollback (cancel) the updates made by PVD, Load Parameters, Load Encrypted Key, and / or Generate PSD Key made after the Transaction Start event was processed.
    - Wipe PSD: Causes PSD to zeroize all plaintext cryptographic keys and CSPs. Used in the remanufacturing process, or to 'clean' the PSD to retry configuration from scratch. This command zeroizes the Unique PSD *Key Encryption Key* (KEK) which results in the loss of all other Private and Secret Keys.

- **Load Vendor Key:** Instructs the PSD to load the (ECDSA-P256) *Vendor Key* Certificate. The PSD verifies the certificate with the *Manufacturing Key*. If valid, PSD stores the certificate, otherwise an error message is generated.
- **Process Audit Response***:* Instructs the PSD to process the *Horizon Audit Response Block* returned from the Pitney Bowes infrastructure. The Audit response must correspond to the immediate previous Audit request command. The PSD verifies the Horizon Audit Response Block with the *Certificate Key*.
- **Process Postage Value Download***:* Instructs the PSD to perform a postage value download operation. The PSD will validate the signature of the *Horizon PVD Block* with the *Certificate Key*.

  Successful verification of this command results in updating the PSD financial registers
- **Process Withdraw Response***:* Instructs the PSD to complete the withdraw process. The PSD verifies the *Horizon Withdraw Response Block* using the *Certificate Key*. If the *Withdraw Response Block* is valid, the PSD removes the funds from the funds registers and sets the state to the Withdrawn State.

  PSD returns the signed Withdraw Certificate in response to this message, if so configured.

User or User (C+):

The User role is utilized by the host device (i.e. Digital Postage Meter). Both User roles have identical services, but the authentication method used depends on the host device.

- **Audit Request:** Instructs the PSD to prepare a signed *Audit Request Block*. The Audit Request Block contains the PSD register values and real time clock value. The record is signed by the *Operation Private Key*.
- **Clear Upload Interval:** Instructs the PSD to clear the Upload Interval Timer.
- **Create Debit Certificate***:* Instructs the PSD to create a debit certificate in the format defined by the Flex Debit Certificate Template.
- **Create PVD Request:** Instructs the PSD to create a *Postage Value Download Request Block*. This contains the current PSD register values and the requested postage amount. It is signed by the *Operation Private Key*.
- **Finalize Debit:** Performs post-debit housekeeping and prepare for the next Debit operation by precomputing the 'r' signature parameter if necessary
- **Log Permit:** Logs the permit and the data capture recovery information.
- **Login Request:** Authenticates the User with the PSD (not applicable for User (C+)). If the authentication is successful, the PSD allows debit operations.
- **Precompute r for Debit:** Pre-computes the 'r' signature component for the PSD Key signature (DSA or ECDSA). This message is used for countries whose debit certificate is signed by a DSA or ECDSA key.
- **Process Flex Debit Block:** Loads a flex debit template into the PSD. The flex debit template defines the indicia content for debit operations. The PSD verifies the signature on the flex debit template using the *Download Key*. If the signature is valid, the flex template is stored in internal memory and used for subsequent debit requests.
- **Sign Transaction Data:** Generates a signature on the included hash. The PSD returns a Transaction Package Hash Block containing the SHA-256 hash digest signed by the *Operation Private Key*.
- **Verify Hash Block**: Validates the included hash. The message contains a hash field and a *Horizon Binary Hash Block*. The PSD validates the signature on the *Horizon Binary Hash Block* using the *Certificate Key*. If the block is valid, the PSD will compare the hash passed in the message with the hash embedded in the *Horizon Binary Hash Block*. If both the signature verification and the hash compare are successful, the PSD returns a Success status.
- **Verify Mail Piece Data**: Verifies the hash of the transaction data for a mail piece. The PSD verifies the signature using the *PSD Mail Piece Key* and returns the hash and the status of the verification.
- **Withdraw Request:** Instructs the PSD to initiate a *Withdrawal* operation. The PSD will enter a locked state (*Withdrawal Pending*) that will not permit any financial operations. The PSD creates a *Withdraw Request Block* containing the PSD's register values. The PSD signs the *Withdraw Request Block* with the *Operation Private Key*.

  The only way to exit the locked state is by completing the withdraw process or by the Data Center aborting the withdraw operation in a Withdraw Response block.

Unauthenticated Services:

Miscellaneous functions that do not require the PSD authentication of the entity. Unauthenticated Services are available to all roles, both authenticated and unauthenticated.

- **Get Challenge:** Returns an 8-byte nonce (random number) from the DRBG, which is used in a subsequent command that requires that nonce word for authentication. This is always done in conjunction with another authorized transaction and is then considered as being done on behalf of any role that requires a nonce value.
- **Get Clock Offsets:** Returns the drift and GMT offset values.
- **Get Flex Debit Template:** Returns the loaded flex debit template.
- **Get GMT Time:** Returns the real time clock value with only the drift correction applied.
- **Get Key List:** Returns a list of all active keys stored in the PSD.
- **Get Local Time:** Returns the real time clock with drift and GMT offsets applied.
- **Get ML Attributes:** Returns device versions and unique device serial number.
- **Get Parameters:** Returns parameter values stored in the PSD. The Host can request individual parameter IDs or all the Parameters in the PSD.
- **Get PSD Attributes:** Returns PSD attribute data, including firmware and hardware versions.
- **Get PSD Status:** Returns PSD status information that includes the module's mode of operation indicator.
- **Get Withdraw Certificate:** Retrieves the *Withdraw Certificate* created at the successful completion of the Withdraw process.
- **Perform Diagnostic Test:** The User sends this message to request that the PSD perform a diagnostic test.
- **Perform Full Diagnostics:** The User sends this command to request the PSD perform its diagnostic processing. The PSD will run its power up tests as well perform other maintenance activities.
- **Read Log File:** Returns Log Data stored in the PSD. The number of available entries, the size of each entry, and the data contained in each entry will depend on the type of log requested.
- **Reboot PSD:** Restarts the PSD application.
- **Set Clock:** Sets the real time clock in the PSD. The real time clock can only be set when the PSD is in manufacturing state. It cannot be changed once the PSD is 'Locked'. The real time clock is set to GMT.
- **Set GMT Offset:** Sets the GMT offset in the PSD. The GMT offset is a combination of time zone offset and daylight savings time offset (if applicable).

3.2.1 Service Access to Security Functions and CSPs

Critical Security Parameter (CSP) and Public Security Parameter (PSP) access by services is classified as Read (R), Write (W), or Zeroize (Z) in the table below.

Table 5 – Services Available in FIPS Approved Mode

Role(s) with Service Access	Service	Security Functions Used	CSP Access	PSP Access
Crypto Officer	Generate PSD Key	ECDSA P-256/P-224 KeyGen Or DSA 2048 KeyGen DRBG AES 256 HMAC-SHA-256	Operation Private Key: W Or Debit Private Key: W DRBG Working State: R, W KEK: R KAK: R	Vendor Key
	Generate Session Key	DRBG SP 800-56A KAS-SSC SP 800-56C KDF ECDSA P-256 SigVer AES KW 256 HMAC-SHA-256 Or AES 256	DRBG Working State: R, W Shared Secret: R, W ECC-CDH PSD KAS Key: R, W Operation Private Key: R Session Authentication Key: W, R Or Session Privacy Key: W, R KEK: R KAK: R	Certificate Key: R ECC-CDH Base KAS Public Key: R, W ECC-CDH Infrastructure KAS Public Key: R, W ECC-CDH PSD KAS Public Key: R, W
	Load Certificate Key	HMAC-SHA-256 ECDSA P-256 SigVer	KAK: R	Vendor Key: R Certificate Key: W
	Load CRL	ECDSA P-256 SigVer	None	Download Key: R
	Load Download Key	HMAC-SHA-256 ECDSA P-256 SigVer	KAK: R	Certificate Key: R Download Key: W
	Load Encrypted Key	HMAC-SHA-256 AES KW 256 AES 256 ECDSA P-256 SigVer	Debit Secret Key: W Session Privacy Key: R KEK: R KAK: R	Certificate Key: R
	Load Key Acknowledgement	ECDSA P-256 SigVer	None	Certificate Key: R
	Load Parameters:			
	Transition to Operational State	ECDSA P-256 SigVer	None	Certificate Key: R
	Transition to Base State	ECDSA P-256 SigVer	None	Certificate Key: R
	Disable PSD	ECDSA P-256 SigVer	None	Certificate Key: R
	Enable PSD	ECDSA P-256 SigVer	None	Certificate Key: R
	Reinitialize PSD	ECDSA P-256 SigVer	KEK: Z	Certificate Key: R
	Start Software Update	ECDSA P-256 SigVer	None	Certificate Key: R, SWAK
	Transaction Start	ECDSA P-256 SigVer	None	Certificate Key: R
	Transaction Commit	ECDSA P-256 SigVer	None	Certificate Key: R
	Transaction Rollback	ECDSA P-256 SigVer	None	Certificate Key: R
	Wipe PSD	ECDSA P-256 SigVer	KEK: Z	Certificate Key: R

Role(s) with Service Access	Service	Security Functions Used	CSP Access	PSP Access
	Load Vendor Key	HMAC-SHA-256 ECDSA P-256 SigVer	KAK: R	Manufacturing Key: R Vendor Key: W
	Process Audit Response	ECDSA P-256 SigVer	None	Certificate Key: R
	Process Postage Value Download	ECDSA P-256 SigVer	None	Certificate Key: R
	Process Withdraw Response	ECDSA P-256/P-224 SigVer	Debit Private Key: R	Certificate Key: R
User/User (C+)	Audit Request	DRBG AES 256 ECDSA P-256 SigGen	DRBG Working State: R, W KEK: R Operation Key: R	None
	Clear Upload Interval	None	None	None
	Create Debit Certificate	DRBG AES 256 ECDSA P-256/P-224 SigGen Or DSA 2048 SigGen Or HMAC-SHA-256	DRBG Working State: R, W KEK: R Debit Secret Key: R Or Debit Private Key: R Or Mail Piece Key: R	None
	Create PVD Request	DRBG AES 256 ECDSA P-256 SigGen	DRBG Working State: R, W KEK: R Operation Key: R	None
	Finalize Debit	None	None	None
	Log Permit	None	None	None
	Login Request (User only)	AES 256	Password: R KEK: R	None
	Precompute r for Debit	DRBG AES 256	DRBG Working State: R, W KEK: R	None
	Process Flex Debit Block	ECDSA P-256 SigVer	None	Download Key: R
	Sign Transaction Data	DRBG AES 256 ECDSA P-256 SigGen	DRBG Working State: R, W KEK: R Operation Key: R	None
	Verify Hash Block	ECDSA P-256 SigVer	None	Download Key: R
	Verify Mail Piece Data	HMAC-SHA-256	Mail Piece Key: R	None
	Withdraw Request	ECDSA P-256 SigGen	Operation Key: R	None
Unauthenticated Role	Get Challenge	DRBG AES 256	DRBG Working State: R, W KEK:R	None
	Get Clock Offsets	None	None	None
	Get Flex Debit Template	None	None	None
	Get GMT Time	None	None	None
	Get Key List	None	None	None
	Get Local Time	None	None	None

Role(s) with Service Access	Service	Security Functions Used	CSP Access	PSP Access
	Get ML Attributes	None	None	None
	Get Parameters	None	None	None
	Get PSD Attributes	None	None	None
	Get PSD Status	None	None	None
	Get Withdraw Certificate	None	None	None
	Perform Diagnostic Test	None	None	None
	Perform Full Diagnostics	None	None	None
	Read Log File	None	None	None
	Reboot PSD	None	None	None
	Set Clock	None	None	None
	Set GMT Offset	None	None	None

3.3 Non-Approved Mode Roles and Services

The non-Approved Mode of the module implements the same roles and services as the Approved Mode of operations, but this mode also allows the use of the algorithms specified in Section 7.3 FIPS Non-Approved Algorithms. Additionally, non-Approved mode includes the following service:

- **Generate Finalizing Franking Record**: A User role sends this command to request that the PSD prepare a signed Finalizing Franking Record. This message includes a hash implemented according to the Germany FrankIt specification. The IndiciaSecurityType parameter must be set to Germany FrankIt. Data items include Indicia Serial Number, ascending register, descending register, piece count, and other defined data items.

3.4 Security Rules

This section documents the security rules enforced by the cryptographic module to implement the security requirements of a 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 or user ID/password.
- All keys generated in the module shall have at least 112 bits of cryptographic security strength for an Approved mode of operation.
- All methods of key generation shall be at least as strong as the key being generated.
- Signed digital indicium data shall not be output unless the proper funds accounting has been performed.
- The module shall not provide a bypass state where plaintext information is 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 shared keys from the module.
- Keys shall be established 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.

4. PHYSICAL SECURITY

The X4i PSD utilizes the Maxim Semi-Conductor MAX32590 micro-controller, a single chip, cryptographic module that protects key material from unauthorized disclosure, modification or substitution. The module is conformant to FIPS 140-2 Level 3 physical security requirements and is protected by an encapsulate. The hardness of the module encapsulate was tested at room temperature and over the module's documented operating temperature range from -40˚C to + 85˚C.

In addition to Level 3 physical security features, the module includes real time environmental monitoring (temperature, battery, voltage), and tamper detection and response. Triggering the environmental failure protection mechanisms or damaging the active shield (tamper detection) that protects the entire module results in a tamper event. A tamper event halts the processor and automatically zeroizes the master key encryption key (KEK).

The operator should periodically inspect the module for evidence of tampering.

5. MITIGATION OF OTHER ATTACKS

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 a Level 4 module. The module is designed to defend against out of bound voltage and temperature extremes. Additionally, the module provides a tamper detection and response mechanism.

6. 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.

7. CRYPTOGRAPHIC KEY MANAGEMENT

7.1 FIPS Approved Algorithms

The following FIPS Approved cryptographic algorithms listed in Table 6 are supported by the module.

Table 6 – FIPS Approved Algorithms

CAVP Certs	Algorithm	Standards	Modes/ Methods	Key Lengths, Curves, or Moduli	Use
5954	AES	FIPS 197 SP 800-38A SP 800-38F	CBC, ECB, KW	256[2]	Data encryption and decryption Cryptographic key wrapping and unwrapping (KTS key establishment providing 256 bits of encryption)
Vendor Affirmed	CKG	SP 800-133			Symmetric key generation and asymmetric seed generation from the unmodified output of the DRBG
C472	DRBG	SP 800-90A	HASH-based		Deterministic Random Bit Generator with 256-bit security-strength (seeded with full entropy). DRBG does not support reseed.
C475	DSA	FIPS 186-4	KeyGen SigGen SigVer	(2048, 224)[3] (2048, 256) (2048, 224, SHA-224) (2048, 256, SHA-256) (2048, 224, SHA-224) (2048, 256, SHA-256)	Generation of cryptographic key pairs, and digital signature generation and verification.
C476	ECDSA	FIPS 186-4	KeyGen SigGen SigVer	P-224[4] P-256 P-224, SHA-256 P-256, SHA-256 P-224, SHA-256 P-256, SHA-256	Generation of cryptographic key pairs, and digital signature generation and verification.

[2] Key sizes 128 and 192 are included in the algorithm certificate, but are not used in Approved mode

[3] The following DSA functionality is included in the algorithm certificate, but is not used in Approved mode: SigVer (1024, 160, SHA-1)

[4] The following ECDSA functionality is included in the algorithm certificate, but is not used in Approved mode: SigGen Component; SigVer (P-192, SHA-1)

CAVP Certs	Algorithm	Standards	Modes/ Methods	Key Lengths, Curves, or Moduli	Use
C464	HMAC	FIPS 198-1	HMAC-SHA-256[5]	256 bits	Used to generate Message Authentication Codes (MACs). Truncated MACs (at least 64 bits) are used for some applications.
Vendor Affirmed	KAS-SSC	SP 800-56Ar3	ECC	P-256	Key Agreement Protocol used to establish a session key (Ephemeral Unified Model C (2e, 0s, ECC CDH))
Vendor Affirmed	KDA	SP 800-56Cr1	One-Step KDF	SHA-256	Key Derivation Function used with KAS-SSC to establish a session key
KTS (AES Cert. #5954)		SP 800-38F	AES KW	256	Protects exported keys (using Session Privacy Key)
KTS (AES Cert. #5954 and HMAC Cert. #C464)		SP 800-38F	AES CBC 256 HMAC-SHA-256	256	Protects CSPs stored in Non-Volatile Memory (NVM) external to the module (using KEK and KAK keys)
C477	RSA	FIPS 186-4	SigVer PKCSPSS	2048, SHA-256[6]	Used for PB Bootloader firmware integrity test
C295	SHS	FIPS 180-4	SHA-224 SHA-256[7]		SHS provides the hashing algorithm necessary for DSA, ECDSA and RSA digital signature generation/ verification and for the key derivation function

7.2 FIPS Allowed Algorithms

The module supports the following non-Approved but Allowed security functions listed in Table 7.

Table 7 – FIPS Allowed Algorithms

Algorithm	Strength	Use
NDRNG	The NDRNG entropy rate and the DRBG implementation ensure that the DRBG is seeded with full entropy (256 bits)	Seeding the DRBG

[5] HMAC-SHA-1 is included in the algorithm certificate, but is not used in Approved mode

[6] The following RSA functionality is included in the algorithm certificate, but is not used in Approved mode: KeyGen (2048); SigGenANSI X9.31 (2048, SHA-256); SigGen PKCS 1.5 (2048, SHA-256); SigGen PKCSPSS (2048, SHA-256); SigVer ANSI X9.31 (1024, SHA-1 and SHA-256) and (2048, SHA-256); SigVer PKCS 1.5 (1024, SHA-1 and SHA-256) and (2048, SHA-256); SigVer PKCSPSS (1024, SHA-1 and SHA-256) and (2048, SHA-1)

[7] SHA-1 is included in the algorithm certificate, but is not used in Approved mode

7.3 FIPS Non-Approved Algorithms

The following cryptographic algorithms listed in Table 8 are used solely in a non-Approved mode of operation (this includes specified CAVP-validated algorithms). There exists no mechanism to allow the use of these algorithms in an Approved mode of operation.

Table 8 – FIPS Non-Approved Algorithms

Algorithm	Key Lengths, Curves, or Moduli	Use
KAS (non-compliant)	1024	FFC KAS used to establish a Triple DES session key
DSA (non-compliant)	KeyGen: (1024, 160) SigGen: (1024, 160, SHA-1) SigVer: (1024, 160, SHA-1)	Used to generate key pairs and generate/verify digital signatures. Legacy verification validated by CAVP DSA Cert. #475.
ECDSA (non-compliant)	KeyGen: P-160 P-192 SigGen: P-160, SHA-1 P-192, SHA-1 SigVer: P-160, SHA-1 P-192, SHA-1	Used to generate key pairs and generate/verify digital signatures. Legacy verification of P-192, SHA-1 validated by CAVP Cert. #C476.
HMAC (non-compliant)	HMAC-SHA-1	Validated by CAVP Cert. #C464
RSA (non-compliant)	KeyGen: 1024 SigGen ANSI X9.31, PKCS 1.5, PKCSPSS: (1024, SHA-1) SigVer ANSI X9.31, PKCS 1.5, PKCSPSS: (1024, SHA-1)	Used to generate keys and digital signatures. Legacy verification validated by CAVP Cert. #C477
SHS (non-compliant)	SHA-1	Hashing for digital signatures and key derivation. Validated by CAVP Cert. #C295
Triple-DES (non-compliant)	2 key and 3 key encrypt/decrypt	Data encryption and decryption. Validated by CAVP TDES Cert. #2900.
Triple-DES MAC (non-compliant)	128-bit, 192-bit	Used to generate Message Authentication Codes (MACs).

7.4 CSPs and Keys

7.4.1 Critical Security Parameters

All CSPs except the KEK are stored in battery-backed memory encrypted by the KEK or in Non-Volatile Memory (NVM) external to the module encrypted by the KEK and protected by the KAK. Therefore zeroizing the KEK destroys access to all CSPs. All CSPs that are input or output are wrapped in conformance to SP 800-38F: CSPs stored in NVM are protected by the KEK and the KAK, while the Debit Secret Key (the only other CSP entered or output) is entered wrapped by the Session Privacy Key.

Table 9 – Secret Keys, Private Keys, Cryptographic Key Components, and Other CSPs

CSP/Key	Security Function	Use	Establishment	Entry/Output	Storage	Destruction
KEK (256-bit) (Key Encryption Key)	AES CBC 256 (Cert. #5954)	Protect all keys stored internally or in NVM	Generated Internally by FIPS approved DRBG (during manufacturing)	Entry: N/A Output: N/A	Plaintext	Zeroization, Tamper or removal of all power
KEK' (256-bit) (Backup Key Encryption Key)	AES CBC 256 (Cert. #5954)	Backup KEK	Generated Internally by FIPS approved DRBG (during manufacturing)	Entry: N/A Output: N/A	Encrypted with KEK	Zeroization, Tamper or removal of all power
KAK (256-bit) (Key Authentication Key)	HMAC-SHA-256 (Cert. #C464)	Protect keys externally stored in NVM	Generated Internally by FIPS approved DRBG (during manufacturing)	Entry: N/A Output: N/A	Encrypted with KEK	Zeroization, Tamper or removal of all power
Debit Private Key (128-bit or 112-bit)	ECDSA P-256 OR ECDSA P-224 (Cert. #C476) OR DSA 2048 (Cert. #C475)	Digitally sign debit records (indicia data)	Generated Internally by FIPS approved DRBG	Entry: N/A Output: N/A	Encrypted with KEK	Zeroization, Tamper or removal of all power
Debit Secret Key (256-bit)	HMAC-SHA-256 (Cert. #C464)	Digitally authenticate debit records (indicia data)	Generated Externally	Entry: Encrypted by Session Privacy Key Output: N/A	Encrypted with KEK	Zeroization, Tamper or removal of all power
Operation Private Key (128-bit)	ECDSA P-256 (Cert. #C476)	Authenticate to the communicating infrastructure	Generated Internally by FIPS approved DRBG	Entry: N/A Output: N/A	Encrypted with KEK	Zeroization, Tamper or removal of all power
Session Authentication Key (256-bit)	HMAC-SHA-256 (Cert. #C464)	Used to authenticate messages sent between the Host and the PSD (User (C+) authentication)	KAS-SSC per SP 800-56Ar3 and KDF per SP 800-56Cr1	Entry: N/A Output: N/A	Encrypted with KEK	Zeroization, Tamper or removal of all power
Session Privacy Key (256-bit)	AES KW 256 (Cert. #5954)	Encrypt data or wrap keys transported to infrastructure	KAS-SSC per SP 800-56Ar3 and KDF per SP 800-56Cr1	Entry: N/A Output: N/A	Encrypted with KEK	Zeroization, Tamper or removal of all power
ECC-CDH PSD KAS Key (256-bit)	KAS-SSC (Vendor Affirmed)	Ephemeral ECC-CDH private key used in KAS	Generated Internally by FIPS approved DRBG	Entry: N/A Output: N/A	Destroyed immediately after session established	Zeroization, Tamper or removal of all power

CSP/Key	Security Function	Use	Establishment	Entry/Output	Storage	Destruction
Shared Secret (256-bit)	KAS-SSC (Vendor Affirmed)	Used to derive session keys	KAS-SSC per SP 800-56Ar3	Entry: N/A Output: N/A	Destroyed immediately after session established	Zeroization, Tamper or removal of all power
DRBG Seed (440-bit)	DRBG Seed	Seeding the DRBG	Generated Internally by NDRNG (during manufacturing)	Entry: N/A Output: N/A	Encrypted with KEK	Zeroization, Tamper or removal of all power
DRBG Working State (1024-bit)	DRBG Working State	Internal working state of the DRBG	Generated Internally by DRBG	Entry: N/A Output: N/A	Encrypted with KEK	Zeroization, Tamper or removal of all power
Mail Piece Key (256-bit)	HMAC-SHA-256 (Cert. #C464)	Authenticate stored mail piece data	Generated Internally by FIPS approved DRBG	Entry: N/A Output: N/A	Encrypted with KEK	Zeroization, Tamper or removal of all power
Password (128-bit)	User Role Password	Authenticate User Role	Externally (during manufacturing)	Entry: N/A Output: N/A	Encrypted with KEK	Zeroization, Tamper or removal of all power

7.4.2 Public Security Parameters Keys

Table 10 – Public Security Parameters

Public Key	Description	Use	Establishment	Entry/Output	Storage
CRK	Customer Root Key (RSA PSS 2048)	Validates the PB Bootloader firmware integrity on power-on	Loaded in Manufacturing	Entry: N/A Output: N/A	Plaintext
SWAK	Software Authentication Key (ECDSA P-256)	Validates the PSD Application firmware integrity on power-on	Loaded in Manufacturing	Entry: N/A Output: N/A	Plaintext
Manufacturing Key	ECDSA P-256 Public	Validates Vendor Certificate	Loaded in Manufacturing	Entry: N/A Output: N/A	Plaintext
Vendor Key	ECDSA P-256 Public	Authenticates CO role	Externally (Loaded)	Entry: Authenticated by Manufacturing Key Output: N/A	Plaintext
Certificate Key	ECDSA P-256 Public	Authenticates CO role. Validates Authority Data, including other public keys	Externally (Loaded)	Entry: Authenticated by Vendor Key Output: N/A	Plaintext
Download Key	ECDSA P-256 Public	Authenticates CO role	Externally (Loaded)	Entry: Authenticated by Certificate Key Output: N/A	Plaintext
ECC-CDH Base KAS Public Key	ECC-CDH KAS Public counterpart received during the DH handshake	ECDH public counterpart received as part of the EC DH exchange.	Externally (Loaded during KAS)	Entry: Authenticated per 56A Output: N/A	Plaintext

Public Key	Description	Use	Establishment	Entry/Output	Storage
ECC-CDH Infrastructure KAS Public Key	ECC-CDH KAS Public counterpart received during the DH handshake	ECDH public counterpart received as part of the EC DH exchange.	Externally (Loaded during KAS)	Entry: Authenticated per 56A and with Certificate Key Output: N/A	Plaintext
ECC-CDH PSD KAS Public Key	ECC-CDH KAS Public key generated during the DH handshake	ECDH public key transmitted as part of the EC DH exchange	Generated Internally	Entry: N/A Output: Plaintext	Plaintext
Debit Key	ECDSA P-256 OR ECDSA P-224 Public OR DSA 2048 Public	Output to the CO. Used to allow the CO to authenticate the debit records	Generated Internally by FIPS approved DRBG	Entry: N/A Output: Signed by Operation Private Key	Plaintext
Operation Key	ECDSA P-256 Public	Output to the CO. Used to allow the CO to authenticate the PSD	Generated Internally by FIPS approved DRBG	Entry: N/A Output: Signed by Operation Private Key	Plaintext

7.4.3 Zeroization

The module is a single-chip, cryptographic module that incorporates an Active Shield that provides a tamper detection and response mechanism. When this mechanism is triggered, the module immediately zeroizes the KEK, which renders all encrypted keys non-operational. The module transitions to a hard error state in which it must be returned to manufacturing.

Zeroization of the module can also be performed by the operator via the *Reinitialize PSD* or *Wipe PSD* services.

8. SELF-TESTS

The module supports the following self-tests.

Power on self-tests (POSTs) can be run on demand by an unauthenticated operator by either power-cycling the module or via the *Reboot PSD* service. Additionally, they may be executed via *Perform Diagnostic Test* or *Perform Full Diagnostics* services.

Upon the failure of any of the self-tests the module transitions to an error state. All data output via the data output interface is inhibited while in the error state. No cryptographic operations can be performed while in the error state. To transition from the error state the module must be power-cycled.

8.1 Power on Self-Tests

Firmware Integrity Tests:
The module conducts the following digital signature verifications on power-up.

- PB Bootloader
  - RSA 2048 (Cert. #C477) Digital Signature Verification
- PSD Application

    - ECDSA P-256 (Cert. #C476) Digital Signature Verification

Algorithm Tests:
The module conducts the following Known Answer Tests (KATs) and Pairwise Consistency Tests (PWCT) on power-up.

- AES (Cert. #5954)
    - AES-256 ECB Encrypt KAT
    - AES-256 ECB Decrypt KAT
- DRBG (Cert. #C472)[8]
    - Instantiate KAT
    - Generate KAT
- DSA (Cert. #C475)
    - 2048 Signature Generation and Verification PWCT
- ECDSA (Cert. #C476)
    - P-256 Signature Generation and Verification PWCT
- HMAC (Cert. #C464) and SHS (Cert. #C295)
    - HMAC-SHA-256 KAT
- Key Agreement: KAS-SSC (vendor affirmed, C(2e, 0s, ECC CDH)) and KDA (vendor affirmed)
    - Primitive "Z" Computation KAT
    - KDF KAT covered by HMAC-SHA-256 KAT

Critical Function Tests:

- RTC Test
- BRAM Pattern Test

8.2 Conditional Tests

The module conducts the following conditional tests.

- NDRNG
    - Repetition Count Test (per IG 9.8)
- PWCT upon cryptographic key pair generation:
    - DSA 2048 PWCT
    - ECDSA P-256 PWCT
- KAS-SSC (vendor affirmed, C(2e, 0s, ECC CDH))
    - ECC Full Public Key Validation per SP 800-56Arev 3: 5.6.2.3.3
- Software/Firmware Load Test:
    - ECDSA P-256 Signature Verification (Cert. #C476)

[8] Per IG 9.8, the SP 800-90A-compliant DRBG does not perform the test described in AS.09.42 and AS.09.43

APPENDIX A: REFERENCES

Table 11 – References

Reference Title	Publishing Entity	Publication Date
Digital Signature Standard (DSA) – FIPS PUB 186-4	NIST	July 2013
Advanced Encryption Standard (AES) – FIPS PUB 197	NIST	November 2001
The Keyed-Hash Message Authentication Code (HMAC) – FIPS PUB 198-1	NIST	July 2008
Secure Hash Standard – FIPS PUB 180-4	NIST	March 2012
Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography - Special Publication 800-56A Revision 3	NIST	April 2018
Recommendation for Key-Derivation Methods in Key-Establishment Schemes - Special Publication 800-56C Revision 1	NIST	April 2018
Recommendation for Block Cipher Modes of Operation, Methods and Techniques – Special Publication 800-38A	NIST	December 2001
Recommendation for Random Number Generation Using Deterministic Random Bit Generators – Special Publication 800-90A	NIST	January 2012
FIPS PUB 140-2, Security Requirements for Cryptographic Modules	NIST	May 2001
Derived Test Requirements for FIPS PUB 140-2, Security Requirements for Cryptographic Modules	NIST	January 2011
FIPS PUB 140-2, Annex A – Approved Security Functions for FIPS PUB 140-2	NIST	January 2018
FIPS PUB 140-2, Annex B – Approved Protection Profiles for FIPS PUB 140-2	NIST	December 2016
FIPS PUB 140-2, Annex C – Approved Random Number Generators for FIPS PUB 140-2	NIST	January 2016
FIPS PUB 140-2, Annex D – Approved Key Establishment Techniques for FIPS PUB 140-2	NIST	May 2018
Implementation Guidance for FIPS PUB 140-2 and the Cryptographic Module Validation Program	NIST	August 2019

APPENDIX B: ABBREVIATIONS AND DEFINITIONS

Table 12 – Abbreviations and Definitions

Term	Definition
AES	Advanced Encryption Standard
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
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
FRDIs	Funds Relevant Data Items
HMAC	Hashed Message Authentication Code
KAS	Key Agreement Scheme
NDRNG	Non-Deterministic Random Number Generator
NVM	Non-Volatile Memory
PB	Pitney Bowes
POST	Power-on Self-test
PSD	Postal Security Device
PSS	Probabilistic Signature Scheme
PVD	Postage Value Download
RAM	Random Access Memory
ROM	Read-Only Memory

Term	Definition
RSA	Rivest Shamir Adleman
RTC	Real Time Clock
SDU	Software Download Utility
SHA	Secure Hash Algorithm
SRAM	Static RAM