# SPYCOS® 3.0 QFN FIPS 140-2 Non-Proprietary Security Policy

Revision: 1.0

**Trademarks**
SPYRUS, the SPYRUS logos, SPYCOS, Rosetta, Rosetta Micro®, are either registered trademarks or trademarks of SPYRUS, Inc. in the United States and/or other countries.

All other trademarks are the property of their respective owners.

**Patents**
Rosetta Authentication Products including SPYCOS®, Rosetta Micro®, the **Rosetta® Series II Smart Cards and USB Security Devices, Rosetta SDHC™ Card, Rosetta MicroSD Memory Card,** may be covered by one or more of the following patents: U.S. Patent No. 6,088,802 and U.S. Pat. No. 6,981,149.

# Contents

# 1   Introduction

This Security Policy specifies the security rules under which the SPYCOS® 3.0 QFN operates.  The Acronym SPYCOS stands for "SPYRUS Cryptographic Operating System". SPYCOS® 3.0 QFN conforms to FIPS 140-2 Security Requirements For Cryptographic Modules.

Included in these rules are those derived from the security requirements of FIPS 140-2 and additionally, those imposed by SPYRUS, Inc.  These rules, in total, define the interrelationship between:

1. Operators,
2. Services, and
3. Critical Security Parameters (CSPs).

## 1.1   SPYCOS® 3.0 QFN Overview

The SPYCOS® 3.0 QFN is the latest addition to the SPYRUS family of cryptographic module ICs that enable both smart card and USB cryptographic tokens.

The SPYCOS® 3.0 QFN enables security critical capabilities such as operator authentication, message privacy, integrity, authentication, and non-repudiation; and secure storage, all within a tamper-evident protective coating. The SPYCOS® 3.0 QFN communicates with a host computer via the ports/interfaces defined in Table 2-1 below.

## 1.2   SPYCOS® 3.0 QFN Implementation

The SPYCOS® 3.0 QFN is implemented as a single-chip module as defined by FIPS 140-2.

The SPYCOS® 3.0 QFN is available with a standard interface QFN mounted package with product name: Rosetta Micro®. All Interfaces have been tested and are compliant with FIPS 140-2. Product Identification (including unique part number) for the SPYCOS® 3.0 QFN is shown in the table below:

**Table 1-1  SPYCOS® 3.0 QFN Product Identification**

| Form Factor | Part Number(s) | FW Version |
|---|---|---|
| Rosetta Micro® | 742100004F | 3.0 |

Images of the above form factor is shown in the figure below:
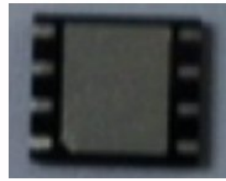


**Figure 1 SPYCOS® 3.0 QFN with Rosetta Micro® Form Factor**

## 1.3   SPYCOS® 3.0 QFN Cryptographic Boundary

The Cryptographic Boundary is defined to be the physical perimeter of the SPYCOS® 3.0 QFN and the potting material it is embedded in (see Figure 1).

No hardware or firmware components that comprise the SPYCOS® 3.0 QFN are excluded from the requirements of FIPS 140-2.

## 1.4   Approved Mode of Operation

The module only operates in an Approved mode of operation.

The SPYCOS® 3.0 QFN Approved mode of operation is comprised of the SPYCOS® 3.0 QFN command set.

Approved mode of operation commands which are successfully completed will return a standard success return code. The Error return codes are dependent upon the cause of the failure. Services available under the Approved mode of operation are detailed in Table 3-1 of this Security Policy.

The SPYCOS® 3.0 QFN supports the following FIPS 140-2 Approved algorithms:

**Table 1-2 SPYCOS® 3.0 QFN Approved Algorithms**

| Approved Algorithms | Certificate # |
|---|---|
| **Encryption & Decryption** | |
| Three-Key Triple-DES | 1772 |
| AES (128-bit, 192-bit, 256-bit key) | 3028 |

| Digital Signatures | |
|---|---|
| ECDSA (sign and verify) [P-256, P-384, P-521] | 578 |
| RSA 2048 (sign and verify) | 1611 |
| **Message Authentication Code** | |
| HMAC (Minimum 112 bit key) | 1913 |
| **Hash** | |
| SHA-1, SHA-224, SHA-256, SHA-384, SHA-512 | 2529 |
| **Key Agreement / Key Establishment** | |
| CVL (ECC CDH) [P-256, P-384, P-521] | 419 |
| KAS [P-256, P-384, P-521] | 52 |
| KTS (AES KW with 128-bit, 192-bit, 256-bit key) | 3115 |
| **Approved Deterministic Random Bit Generator** | |
| SP800-90 DRBG | 658 |

Approved ECDSA (Cert. #578). The Digital Signature will provide between 128-bits to 256-bits of equivalent computational resistance to attack depending upon the size of the curves that are used (P-256, P-384, P-521).

Approved RSA (Cert. #1611). The Digital Signature with a 2048 key size will provide 112 bits of equivalent computational resistance to attack.

Approved SP800-56A, Section 5.7.1.2: ECC CDH Primitive (Cert. #419). The key establishment process will provide between 128-bits to 256-bits of equivalent computational resistance to attack depending upon the size of the ECC CDH curves that are used (P-256, P-384, P-521).

Approved KAS ECC (Cert. #52). The key establishment process will provide between 128-bits to 256-bits of equivalent computational resistance to attack depending upon the size of the keys that are used (P-256, P-384, P-521).

Approved KTS (Cert. #3115; key establishment methodology provides between 128 and 256 bits of encryption strength).

The following services are available as "non-Approved" algorithms but allowed:

**Table 1-3  SPYCOS® 3.0 QFN Non-Approved but allowed Algorithms**

| Algorithms |
|---|
| **RNG** |
| HW NDRNG (Only used for seeding Approved SP800-90 DRBG) |
| **Key Wrap & Unwrap** |
| RSA  (key wrapping; key establishment methodology provides 112 bits of encryption strength) |

## 1.5  FIPS 140-2 Security Levels

The SPYCOS® 3.0 QFN cryptographic module complies with the requirements for FIPS 140-2 validation to the levels defined in Table 1-3.  The FIPS 140-2 overall rating of the SPYCOS® 3.0 QFN is Level 3.

**Table 1-3**
**FIPS 140-2 Certification Levels**

| FIPS 140-2 Category | Level |
|---|---|
| 1.  Cryptographic Module Specification | 3 |
| 2.  Cryptographic Module Ports and Interfaces | 3 |
| 3.  Roles, Services, and Authentication | 3 |
| 4.  Finite State Model | 3 |
| 5.  Physical Security | 3 |
| 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 | N/A |
| Overall Security Level | 3 |

# 2   Ports and Interfaces

The pins form a set of 5 contact points that comprise the only electronic interface with external devices. They are the sole externally visible portion of the microprocessor assembly. There are 5 active pins (VCC, RST, CLK, GND and I/O).  The active pins perform the following functions:

**Table 2-1**
**SPYCOS® 3.0 QFN Pins and Logical Interfaces**

| Pin | Function | FIPS 140-2 Logical Interface |
|---|---|---|
| VCC | Operating voltage | Power Interface |
| RST | Reset input | Control Input |
| CLK | Processor clock input | Control Input |
| GND | Ground | Power Interface |
| I/O | Bi-directional data port | Data Input / Data Output; Control Input; Status Output |

# 3  Roles and Services

The SPYCOS® 3.0 QFN supports two roles, Crypto-officer (CO) and User, and enforces the separation of these roles by restricting the services available to each one.

**Crypto-officer Role**: The Crypto-officer is responsible for initializing the SPYCOS® 3.0 QFN. Before issuing the SPYCOS® 3.0 QFN to an end User, the Crypto-officer initializes the SPYCOS® 3.0 QFN with private keying material and certificate information.  The Crypto-officer cannot use private keys loaded on the module. The SPYCOS® 3.0 QFN validates the Crypto-officer identity before accepting any initialization commands.  The Crypto-officer is also referred to as the Site Security Officer (SSO).

**User Role**: The User role is available after the SPYCOS® 3.0 QFN has been loaded with a User personality.  The User can load, generate and use private keys.

The SPYCOS® 3.0 QFN validates the User identity before access is granted.

## 3.1  Services

The following table (Table 3-1) describes the services provided by the SPYCOS® 3.0 QFN.  The User/SSO column denotes the roles that may execute the service.

**Table 3-1**
**SPYCOS® 3.0 MODULE Services**

| Service | Description | User / SSO |
|---|---|---|
| AES UNWRAPKEY | Supports key export by using the AES unwrap key process to decrypt a wrapped key data block, and then storing it in the internal key register or the key file. | User |
| AES WRAPKEY | Supports key export by using the AES wrap key process to encrypt the internal symmetric key data that is transmitted to the host. | User |
| AUTHENTICATE SECURE CHANNEL | Validates the secure channel between the host and the module. | User, SSO |
| BLOCK PIN | Blocks User PIN access. Resets attempt count for the User PIN to zero and prohibits User PIN logon until an UNBLOCK PIN command is executed by the SSO role. | User, SSO |
| CHANGE PASSWORD | SSO changes either the User PASSWORD Phrase or SSO PASSWORD Phrase. | SSO |
| CHECK PASSWORD | User / SSO Inputs a PASSWORD Phrase to authenticate the SSO or the User. | User, SSO |

| Service | Description | User / SSO |
|---|---|---|
| CREATE | A file of type DF, SF, or EF is created[1] | User, SSO |
| DECRYPT | Performs a decryption process on the input data and sets up the plaintext data for retrieval. Supports multiple modes of decryption for User data. | User |
| DELETE | Deletion of a file, directory. | User, SSO |
| DIRECTORY | Retrieval of directory. | User, SSO |
| ECC GENERATE KEY | Creates an ECC public/private key pair for signing/verifying or transport. | User |
| ECDH COMPUTE SECRET | Generates a shared secret, Z. | User |
| ECDSA SIGN | Computation of a digital signature using the ECDSA algorithm using the hash value. | User |
| ECDSA VERIFY | Performs an ECDSA signature verification. | User |
| ENCRYPT | Performs a symmetric encryption process on the input data and returns the ciphertext data. Supports multiple modes of encryption for User data. Get Response must be issued to retrieve the data. | User |
| ENVELOPE | Sends the APDU commands through the secure channel established previously between the host and the SPYCOS® 3.0 QFN. The session key is generated during the secure channel establishment (See Manage Secure Channel). The encryption mode used is the AES CBC mode. | User |
| EXTEND | Extension of the length of a file or directory. | User, SSO |
| FIPS INFO | Returns a value indicating whether the module is in FIPS Mode (1) or not (0) | User, SSO |
| GENERATE HMAC KEY | Generates an HMAC key and initializes the currently selected file for use with the HMAC commands. | User |
| GENERATE IV | Generates an IV. | User |
| GENERATE RANDOM | Generates a random number and also handles the generation of Initialization Vectors (IVs) and Message Encryption Keys (MEKs). | User |
| GENERATE SYMMETRIC KEY | Used to generate Message Encryption Keys (MEKs). It can also generate random numbers and IVs. | User |
| GET PUBLIC | Retrieves the public key information of an ECC key. | User |
| GET RESPONSE | Retrieval of the module response. | User |
| GET SPYCOS VERSION | Retrieves firmware version of module. | User, SSO |
| GET STATUS | Query on the current status of a File. | User, SSO |
| HASH FINALIZE | Completes the hash operation and returns the hash | User |

[1] Refer to ISO/IEC 7816-4 for definition of file types and file system

| Service | Description | User / SSO |
|---|---|---|
|  | value. |  |
| HASH INITIALIZE | Initializes internal state to prepare for hashing operations. | User |
| HASH PROCESS | Optional function called to hash a block of data when its length is an even multiple of the hash algorithm block size. | User |
| HMAC FINALIZE | Processes any remaining bytes in the message and retrieves the HMAC value. | User |
| HMAC INITIALIZE | Generates a HMAC message authentication code. | User |
| HMAC PROCESS | Processes the message in even multiples of the hash algorithm's block size. | User |
| IMPORT HMAC KEY | Imports an HMAC key and initialize the currently selected file for use with the HMAC commands. | User |
| INIT PIN FILE | Used by SSO during initialization of the module to Init the PIN File. | SSO |
| KDFEXTERNAL | Passes the KDF data to the hash function. | User |
| KDFFINAL | Completes the generation of the key. | User |
| KDFINTERNAL | Passes the KDF data found inside the module to the hash function. | User |
| KDFSTART | Sets up the internal hash engine for hashing the subsequent data. The hash type is determined by the settings in specified input parameters. | User |
| LOAD CRYPTOGRAPHIC DATA | Supports RSA / ECDSA signature verification or RSA Wrap Key operation | User |
| LOAD IV | Load IV. | User |
| LOAD KEY | An overloaded function that performs Load MEK (Message Encryption Key), Load IV, or Delete Key. | User |
| LOAD SECRET | Used by SSO during initialization of the module to load the secret. | User, SSO |
| LOCK | Disables all operations on this file. The file can still be selected and the status information can still be retrieved, but its contents cannot be accessed. | User, SSO |
| MANAGE SECURE CHANNEL | Establishes the secure channel between the host and the SPYCOS® 3.0 QFN module. Specific codes sent by the host initialize and terminate the secure channel. | User |
| READ BINARY | Binary read from a file, given the offset and length. | User, SSO |
| RSA GENERATE KEYPAIR | Creates an RSA key pair to be used for signing/verifying or transport. The User must have created the RSA keying file (with appropriate access controls) prior to issuing the GENERATE command. | User, SSO |
| RSA SIGN DATA | Signing a message or data object using RSA signature | User |
| RSA UNWRAP KEY | Enables completion of public key exchange of a MEK. | User |
| RSA VERIFY SIGNATURE | Verifying an RSA signature on a message. | User |
| RSA WRAP KEY | Invocation of an RSA Key wrap service | User |

| Service | Description | User / SSO |
|---------|-------------|------------|
| SELECT | Setting a current file within a logical channel. | User, SSO |
| SELF TEST | Automatically performed at power-up and can be executed on-demand via power cycling the module | User, SSO |
| SET KEY | Setting one of the 3 key pointers to the key registers to be used for encryption and decryption using the following symmetric encryption algorithms: AES and Three-key Triple-DES. | User |
| UNBLOCK PIN | Used by an SSO to restore User PIN logon access. | SSO |
| UNLOCK | Enable a previously locked file. | User, SSO |
| UPDATE BINARY | Update of the data in the currently selected EF[2] with the data provided. | User, SSO |
| XAUTH ENROLL | Set up the shared symmetric key for used with the challenge and response authentication process. | User, SSO |
| XAUTH EXTERNAL AUTHENTICATION | Submits the encrypted result of the challenge data retrieved from the XAUTH Get Challenge command. | User, SSO |
| XAUTH GET CHALLENGE | Establishes the challenge and response authentication process by first requesting the random challenge for the current session, the resulting challenge data is output to the host to calculate the encrypted response for use in comparison with the XAUTH External Authentication command. | User, SSO |
| ZEROIZE | Zeroization of the module. Performed using DELETE with recursive argument. | User, SSO |

In addition to the services listed above in table 3-1, the following non-security relevant services may be executed while the operator is unauthenticated:

- CREATE
- DELETE
- DIRECTORY
- EXTEND
- FIPS INFO
- GET UNAUTHENTICATED RANDOM
- GET RESPONSE
- GET SPYCOS VERSION
- GET STATUS
- READ BINARY
- SELECT
- SELF TEST
- UPDATE BINARY

---

[2] Refer to ISO/IEC 7816-4 for definition of file types

# 4    Identification and Authentication

## 4.1    Initialization Overview

The SPYCOS® 3.0 QFN is initialized at the factory with a Default SSO PASSWORD Phrase. The SSO (Site Security Officer) must change the default value during logon to make the module ready for initialization. During initialization the module allows the execution of only the commands required to complete the initialization process.

Before a User can access or operate the module, the SSO must initialize it with the User PASSWORD Phrase. The SSO is authorized to log on to the module any time after initialization to change parameters. The module allows 10 consecutive failed SSO logon attempts before it zeroizes all key material and initialization values. In the *zeroized* state, the SSO must use the Default SSO PASSWORD Phrase to log on to the module and must reinitialize all module parameters.

A User must log on to a module to access any on-board cryptographic functions. To log on the User must provide the correct User PASSWORD Phrase. The module allows 10 consecutive failed logon attempts before it blocks the stored User Password. User information stored in the module in non-volatile memory remains resident.

## 4.2    Authentication

The SPYCOS® 3.0 QFN implements identity-based authentication which is accomplished by PIN or Password[3] entry by the operator. On invocation by the operator, the SPYCOS® 3.0 QFN waits for authentication of the User or SSO role by entry of a Password Phrase. There is only one User and one SSO Password allowed per module.   Multiple User and SSO accounts are not permitted. The authentication password strength available for each supported role is indicated in Table 4-1 below.

**Table 4-1 Identification and Authentication Roles and Data**

| Role | Type of Authentication | Authentication Data – (Strength) |
|------|------------------------|----------------------------------|
| **Crypto-officer (SSO)** | Identity-based | Password (6 - 20 Bytes) |
| **User** | Identity-based | Password (6 - 20 Bytes) |

---

[3] The terms PIN and Password and PASSWORD Phrase are used synonymously in this document.

Once a valid PASSWORD Phrase has been accepted the SPYCOS® 3.0 QFN cryptographic services may be accessed. The CHECK PASSWORD command includes either the User PASSWORD Phrase as a parameter (or) the SSO PASSWORD Phrase as a parameter. If successful, either the User or SSO gains access to the module.

The SPYCOS® 3.0 QFN stores the number of logon attempts in non-volatile memory.   The count is reset after every successful entry of a User PASSWORD Phrase by a User and after every successful entry of the SSO PASSWORD Phrase by the SSO.  If the User role fails to logon to the SPYCOS® 3.0 QFN in 10 consecutive attempts, the SPYCOS® 3.0 QFN will zeroize the User PASSWORD Phrase, block all of the User Private Keys and Public Keys, block all of the User Key Registers and disallow User access. The SPYCOS® 3.0 QFN then transitions to a state that is initialized only for the SSO to perform restorative actions. Restorative actions performed by the SSO may include reloading of initialization parameters, unblocking the User PASSWORD Phrase, or zeroization of the module. When the SPYCOS® 3.0 QFN is powered up after a zeroize, it will transition to the Zeroized State, where it will only accept the Default SSO PASSWORD Phrase.  After the Default SSO PASSWORD Phrase has been accepted, the SPYCOS® 3.0 QFN transitions to the Uninitialized State and must be reinitialized, as described in section 6.

## 4.3  Strength of Authentication

The strength of the authentication mechanism conforms to the following specifications in Table 4-2. The calculations are based on the enforced minimum PASSWORD Phrase size of 6 bytes.

**Table 4-2  Strength of Authentication**

| Authentication Mechanism | Strength of Mechanism |
|---|---|
| Single Password-entry attempt / False Acceptance Rate | The probability that a random 6-byte Password-entry (using only 93 keyboard characters[4]) attempt will succeed or a false acceptance will occur is $1.5456185 \times 10^{-12}$.  The requirement for a single–attempt / false acceptance rate of no more than 1 in 1,000,000 (i.e. less than a probability of $10^{-6}$) is therefore met. |
| Multiple Password-entry attempts in one minute | There is a maximum bound of 10 successive failed authentication attempts before zeroization occurs. The probability of a successful attack of multiple attempts in a one minute period is no more than $1.5456185 \times 10^{-11}$ due to the enforced maximum number of logon attempts.  This is less than one in 100,000 (i.e., $1 \times 10^{-5}$), as required. |

---

[4] The character set available for PINs is at least all alphanumeric characters (upper and lower cases) and 31 special keyboard characters comprising the set {~ ! @ # $ % ^ & * ( ) _ + - = { } [ ] | \ : ; " ' < , > . ? /}.

### 4.3.1    Obscuration of Feedback

Feedback of authentication data to an operator is obscured during authentication (e.g., no visible display of characters result when entering a password).  The PASSWORD Phrase value is input to the CHECK PASSWORD command as a parameter by the calling application.  No return code or pointer to a return value that contains the PASSWORD Phrase is provided.

### 4.3.2    Non-weakening Effect of Feedback

Feedback provided to an operator during an attempted authentication shall not weaken the strength of the authentication mechanism.  The only feedback provided by the CHECK PASSWORD command is a return code denoting success or failure of the operation.  This information in no way affects the probability of success or failure in either single or multiple attacks.

### 4.3.3    Generation of Random Numbers

The GENERATE RANDOM command can be invoked only after authentication of the User.  The SP800-90 DRBG algorithm is used for all authenticated RNG calls.

# 5    Key Management

## 5.1   CSP Management

**Table 5-1**
**SPYCOS® 3.0 QFN CSPs**

| CSP Designation | Use |
|---|---|
| ECDSA Private Key | The Private Key of the User employed in Elliptic Curve digital signing operations. |
| EC-keypair | Used in ECC CDH key agreement |
| Hash DRBG Seed | Used only in generating the initial state of the SP800-90 Hash_DRBG |
| HMAC Key | Used to generate HMAC message authentication code. |
| Message Encryption Key (MEK) | AES Key or Three-Key Triple-DES Key for User data encryption/decryption. |
| RSA Private Key[5] | The Private Key of the User employed in RSA digital signing operations or unwrapping keys |

---

[5] RSA key pair use varies with the User application.  Some keys are used for signatures and some for key wrapping.

| CSP Designation | Use |
|---|---|
| Secure Channel Session Key | ECDH / AES key used to encrypt and decrypt PASSWORD data transmitted to the module |
| SSO PASSWORD Phrase | A secret 6 - 20 bytes value used for SSO authentication. |
| User  PASSWORD Phrase | A secret 6 - 20 bytes value used for User authentication. |

## 5.2  Public Key Management Parameters

**Table 5-2**
**SPYCOS® 3.0 QFN Public Key Management Parameters**

| Key Management Parameter | Use |
|---|---|
| ECDSA Public Key | The Public Key of the User employed in Elliptic Curve digital signing operations. |
| RSA Public Key | The Public Key of the User employed in RSA digital signing operations. |

## 5.3  CSP Access Matrix

The following table (Table 5-3) shows the services (see section 3.1) of the SPYCOS® 3.0 QFN, the roles (see section 3) capable of performing the service, the CSPs (see section 5.1) that are accessed by the service and the mode of access (see next paragraph) required for each CSP.  The following convention is used: If only one of the roles applies to the service, that role appears alone.  If both roles may execute the service, then "User, SSO" is indicated.  If either one (but not the other) then "User" or "SSO" is indicated.  In the last option it is a matter of organizational policy which of the roles may execute the service.

Access modes are R (read), W (write) and E (execute).   Destruction is represented as a W.

**Table 5-3**
**SPYCOS® 3.0 QFN Access Matrix**

| Service | User / SSO | Access Type | CSP Access |
|---|---|---|---|
| AES UNWRAPKEY | User | R,E | MEK |
| AES WRAPKEY | User | R,E | MEK |
| AUTHENTICATE SECURE CHANNEL | User, SSO | R,W,E | Secure Channel Session Key |
| BLOCK PIN | User, SSO | E | User PASSWORD Phrase, SSO PASSWORD Phrase |
| CHANGE PASSWORD | SSO | W | User PASSWORD Phrase, |

| Service | User / SSO | Access Type | CSP Access |
|---|---|---|---|
| | | | SSO PASSWORD Phrase |
| CHECK PASSWORD | User, SSO | R | User PASSWORD Phrase, SSO PASSWORD Phrase |
| CREATE | User, SSO | N/A | N/A |
| DECRYPT | User | R | MEK |
| DELETE | User, SSO | N/A | N/A |
| DIRECTORY | User, SSO | N/A | N/A |
| ECC GENERATE KEY | User | W | EC-keypair |
| ECDH COMPUTE SECRET | User | N/A | N/A |
| ECDSA SIGN | User | R | ECDSA Private Key |
| ECDSA VERIFY | User | R | ECDSA Private Key |
| ENCRYPT | User | R | MEK |
| ENVELOPE | User | R,E | Secure Channel Session Key |
| EXTEND | User, SSO | N/A | N/A |
| FIPS INFO | User, SSO | N/A | N/A |
| GENERATE HMAC KEY | User | R,E | HMAC Key |
| GENERATE IV | User | N/A | N/A |
| GENERATE RANDOM | User | R | HASH DRBG Seed |
| GENERATE SYMMETRIC KEY | User, SSO | W | MEK |
| GET PUBLIC | User | N/A | N/A |
| GET RESPONSE | User | N/A | N/A |
| GET SPYCOS VERSION | User, SSO | N/A | N/A |
| GET STATUS | User, SSO | N/A | N/A |
| HASH FINALIZE | User | N/A | N/A |
| HASH INITIALIZE | User | N/A | N/A |
| HASH PROCESS | User | N/A | N/A |
| HMAC FINALIZE | User | W | HMAC Key |
| HMAC INITIALIZE | User | W | HMAC Key |
| HMAC PROCESS | User | W | HMAC Key |
| IMPORT HMAC KEY | User | R,W | HMAC Key |
| INIT PIN FILE | SSO | R,W | User PASSWORD Phrase, SSO PASSWORD Phrase |
| KDFEXTERNAL | User | N/A | N/A |
| KDFFINAL | User | W | Secure Channel Session Key |
| KDFINTERNAL | User | N/A | N/A |
| KDFSTART | User | N/A | N/A |
| LOAD CRYPTOGRAPHIC DATA | User | N/A | N/A |

| Service | User / SSO | Access Type | CSP Access |
|---|---|---|---|
| LOAD IV | See Load Key | N/A | N/A |
| LOAD KEY | User | W | MEK |
| LOAD SECRET | User, SSO | R | User PASSWORD Phrase, SSO PASSWORD Phrase |
| LOCK | User, SSO | N/A | N/A |
| MANAGE SECURE CHANNEL | User, SSO | W | Secure Channel Session Key |
| READ BINARY | User, SSO | N/A | N/A |
| RSA GENERATE KEYPAIR | User, SSO | W | RSA Private Key |
| RSA SIGN DATA | User | R,E | RSA Private Key |
| RSA UNWRAP KEY | User | R R | RSA Private Key MEK |
| RSA VERIFY SIGNATURE | User | R,E | RSA Private Key |
| RSA WRAP KEY | User | R,W | RSA Private Key MEK |
| SELECT | User, SSO | N/A | N/A |
| SELF TEST | User, SSO | N/A | N/A |
| SET KEY | User | N/A | N/A |
| UNBLOCK PIN | SSO | W | User PASSWORD Phrase, SSO PASSWORD Phrase |
| UNLOCK | User, SSO | N/A | N/A |
| UPDATE BINARY | User, SSO | N/A | N/A |
| XAUTH ENROLL | User, SSO | N/A | N/A |
| XAUTH EXTERNAL AUTHENTICATION | User, SSO | N/A | N/A |
| XAUTH GET CHALLENGE | User, SSO | N/A | N/A |
| ZEROIZE | User, SSO | W | ECDSA Private Key EC-keypair Hash DRBG Seed HMAC Key Message Encryption Key (MEK) RSA Private Key Secure Channel Session Key SSO PASSWORD Phrase User PASSWORD Phrase |

## 5.4  Destruction of Keys and CSPs

The module has the ability to destroy all keys and CSPs by a recursive DELETE command.  All keys and CSPs are stored in files.  The contents of the file(s) being recursively deleted are erased and over written.  Should a power-down occur during the execution of the recursive DELETE, the action of zeroization will resume on a subsequent power-on event, ensuring that access to zeroized information is prevented.

# 6      Setup and Initialization

The uninitialized module has only a root directory with minimal version and manufacturing information in specific files.  There is no information pertaining to the User or SSO or their authentication data, such as Passwords, stored on the uninitialized module as shipped to the customer.

Initialization of the module is accomplished by setting up a security domain by following the procedures below:

- The SSO creates a new application directory on the module;
- The SSO creates a PIN file that is associated with the SSO and User;
- The SSO initializes the PIN files;
- The SSO may optionally set a default Password or set the User PASSWORD Phrase:
  - If the User PASSWORD Phrase is set by the SSO, the User will not be able to change their Password.
- The SSO uses FIPS INFO command to confirm FIPS mode.

The module is now in FIPS mode and operators may logon with the CHECK PASSWORD command.  See Section 4.2 for a description of the CHECK PASSWORD process.

# 7    Physical Security

The module is packaged to meet FIPS 140-2 Level 3 Security. The chip is packaged with physical security mechanisms that destroy the chip if physical attacks are launched against it.  This is achieved using a hard, opaque, tamper-evident coating on the chip.

**Table 7-1**
**Inspection of Physical Security Mechanisms**

| Physical Security Mechanisms | Recommended Frequency of Inspections | Inspection/Test Guidance Details |
|---|---|---|
| Hard, opaque, tamper-evident coating on the chip. | As often as feasible, based upon organization security policy. | Inspect the cryptographic boundary for scratches, scrapes, divots and other suspicious markings or indicators of malice and tampering.  If any signs of suspicious activity are observed, return the cryptographic module to SPYRUS. |

# 8   Self-Tests

The module performs both power-on and conditional self-tests.  The power-on self-tests run automatically when power is restored to the module, without requiring any actions or inputs from the operator.

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

- Firmware Integrity Test with 160-bit Error Detection Code
- Cryptographic algorithm known answer tests (KAT) for:
    - Three-key Triple-DES KAT (encrypt)
    - Three-key Triple-DES KAT (decrypt)
    - AES KAT (encrypt)
    - AES KAT (decrypt)
    - ECDSA KAT (sign)
    - ECDSA KAT (verify)
    - ECC CDH KAT
    - RSA KAT (sign)
    - RSA KAT (verify)
    - HMAC (SHA-1, SHA-256, SHA-512) KAT
    - SP800-90 DRBG KAT

Power cycling allows either the User or SSO to perform any or all of the above tests on demand.

The module performs the following conditional tests:

- ECDSA Pairwise Consistency Test
- ECC CDH Pairwise Consistency Test
- RSA Pairwise Consistency Test
- Continuous test for Approved SP800-90 DRBG
- Continuous test for non-Approved NDRNG

# 9   Mitigation of Other Attacks

The module is not claimed to mitigate against any specific attacks.

**Table 9-1**
**Mitigation of Other Attacks**

| Other Attacks | Mitigation Mechanism | Specific limitations |
|---|---|---|
| Not applicable. | Not applicable. | Not applicable. |