DECRU

# DATAFORT SEP

SAN Configuration:
HW PN/Rev: 60-000109/A
FW PN: SAN 29.4
SW PN: 23.3

NAS Configuration:
HW PN/Rev: 60-000109/A
FW PN: NAS 29.4
SW PN: 23.3

# SECURITY POLICY

version 1.4.12, April 19, 2004

# TABLE OF CONTENTS

# 1 INTRODUCTION

The DataFort™ Storage Encryption Processor (SEP) is a multi-chip embedded module that is the main cryptographic service provider for Decru DataFort storage encryption products.

Decru DataFort is an appliance that intercepts data sent between a client machine and storage device; DataFort transparently encrypts data sent to storage, and decrypts data served to the client. Software running on the DataFort platform manages encrypted keys, performs client authentication, access control, and requests cryptographic services from the SEP.

## 1.1 PURPOSE OF THE CRYPTO MODULE

The purpose of the SEP is to:

- Encrypt/decrypt client data using a hardware AES-256 ECB engine.
- Generate keys from an Approved FIPS 186-2 change notice 1 Appendix 3.1 Deterministic RNG system that includes a commercial “true” random number generator for seeding.
- Establish keys using commercially available key establishment protocols as allowed by FIPS PUB 140-2 Annex D.
- Authenticate platform software and other SEP modules.
- Physically protect plaintext cryptographic keys and CSPs.

# 1.2 PHYSICAL EMBODIMENT

The SEP is embedded into the Decru Crypto Card (DCC) and the SEP cryptographic boundary is defined as the outer perimeter of the potted portion of the printed circuit board. The DCC is a PCI Card conformant to the PCI bus 2.0 standard. The DCC also contains additional (non cryptographic) hardware components that are outside of the physically contiguous cryptographic boundary. These components serve as custom add ons for the DataFort platform (for example, battery backed RAM) outside of the cryptographic boundary.

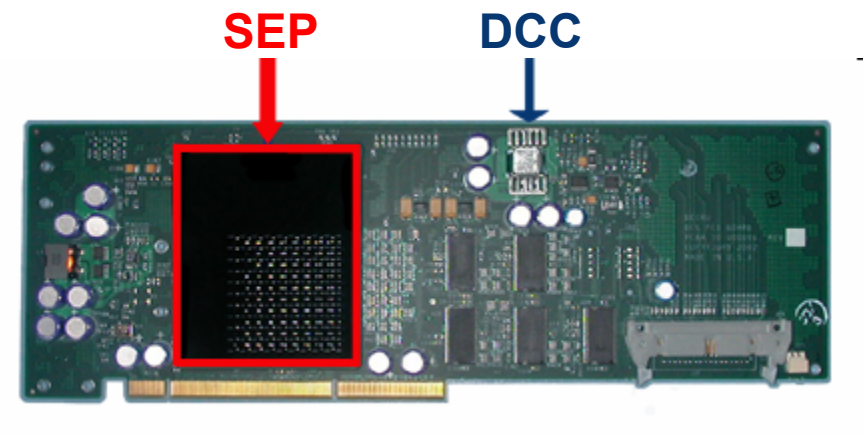


**Figure 1: Module Embodiment (Primary Side)**

Both the primary and secondary sides of the SEP are covered in epoxy potting (secondary side not pictured). Figure 1 depicts the entire DCC card including the SEP cryptographic module (the potted portion of the card included within the red border) and other components that are outside of the cryptographic boundary.

### 1.2.1 CRYPTO MODULE CONFIGURATION

The SEP module as certified has two physical configurations:

| SAN Configuration | NAS Configuration |
|---|---|
| HW PN/Rev: 60-000109/A | HW PN/Rev: 60-000109/A |
| FW PN: SAN 29.4 | FW PN: NAS 29.4 |
| SW PN: 23.3 | SW PN: 23.3 |

The SAN configuration includes interfaces for Dual Data Rate (DDR) RAM that is a DCC add-on component that is outside of the boundary, whereas the NAS configuration does not include this interface.

## 1.3 PORTS AND INTERFACES

This section defines the module's physical ports and the module's primary logical interfaces. The following table maps the module's physical ports to the FIPS interface classification.

**Table 1: Interface Classification**

| Physical Port(s) | FIPS Interface(s) |
|---|---|
| PCI | Status Output, Control Input, Data output/input , Power |
| DDR bus | Data Input, Data Output |
| LCD line | Data Input, Data Output |

| Physical Port(s) | FIPS Interface(s) |
|---|---|
| Tamper line | Control Input |
| I2C | Control Input, Data Output |
| LED bus | Data Output |
| TestPoint bus | Control Input, Status Output |
| Voltage bus | Control Input |
| Backup power | Power |

## 1.4 SECURITY LEVEL

The SEP meets the overall requirements applicable to Level 3 security of FIPS PUB 140-2. The following table lists the compliance level of each section:

**Table 2: Security Level**

| Security Requirements Section | Level |
|---|---|
| Cryptographic Module Specification | 3 |
| Modules, Ports, and Interfaces | 3 |
| Roles, Services, and Authentication | 3 |
| Finite State Model | 3 |
| Physical Security | 3 |

| Security Requirements Section | Level |
|---|---|
| Operational Environment | N/A |
| Cryptographic Key Management | 3 |
| EMI/EMC | 3 |
| Self-Tests | 3 |
| Design Assurance | 3 |
| Mitigation of other attacks | N/A |

# 2 IDENTIFICATION AND AUTHENTICATION POLICY

The SEP supports four authenticated operator roles:

**Table 3: Roles and Authentication**

| Role | Type of Authentication | Authentication Data |
|---|---|---|
| user | Identity-based operator authentication | Message authentication key (20 octet HMAC-SHA-1 key)<br><br>used in an AKEP2 protocol |
| primary cryptographic officer | | |
| cluster officer | Identity-based operator authentication | ECC-521 key pair/16 octet ID<br><br>used in a Needham Schroeder Lowe protocol |
| upgrade firmware | Identity-based operator authentication | ECC-521 ECDSA signature verification key, and the ECDSA signature (using SHA-1) of a firmware upgrade package |

## 2.1 THE SYSTEM USER IDENTITY

The System User assumes the roles of user and primary crypto officer. The System User corresponds to the software controlling the DataFort platform. There may be only a single System User per module, and the ID of this user defaults to a fixed global value.

The System User is authenticated by performing an AKEP2 protocol with the module, which is a commercially available key establishment protocol.

Successful authentication authorizes the System User to assume both the primary crypto officer and user role. However, the System User may elect to relinquish the primary crypto officer role, remaining in user role only. In this state, successful AKEP2 re-authentication is required for the System User to reassume the primary crypto officer role.

The System User is assigned two roles in order to fulfill the security policy of the platform. Specifically, in Decru's platform, authentication key material for the AKEP2 protocol is stored in a smart card that is loaded into the platform at boot, and may be subsequently removed (restricting platform software to the user role.) This allows the platform to be managed within a different security environment than that in which it is deployed for runtime use.

## 2.2 CLUSTER OFFICER IDENTITY

The SEP supports up to 31 cluster officers. Each cluster officer is identified by a unique ID and an ECC-521 key pair. Cluster officer authentication is through the commercially available Needham-Schroeder-Lowe protocol. Cluster officers authenticate in order to access a single service (authentication and key agreement) which is performed in the course of successful authentication. Cluster officers cannot remain authenticated to the module after accessing this service (a prior authentication and key agreement attempt does not influence subsequent authentication and key agreement attempts.)

# 2.3 DECRU IDENTITY

The Decru identity is authorized to assume the upgrade firmware role. Decru's identity is bound to an ECC-521 key pair, used exclusively for ECDSA signatures. The verification key is embedded in the module, and the Decru identity is authenticated via an ECDSA signature verification process, which is part of the upgrade service. Requesting the upgrade service places the SEP in a special state, during which no other authenticated users may access the module.

The following table summarizes the authentication strengths of the three authentication protocols:

**Table 4: Authentication strength**

| Authentication Mechanism | Strength of Mechanism |
|---|---|
| AKEP2 | The odds of successful random authentication are 1 in 2^160. The odds of successful authentication after multiple random attempts in one minute are less than one in 2^146. |
| Needham-Schroeder-Lowe | The odds of a successful random authentication is less than 1 in 2^256. The odds of successful authentication after multiple random attempts in one minute are less than one in 2^243. |
| Firmware signature verification (SHA-1/ECDSA) | The odds of a successful random authentication is less than 1 in 2^80. The odds of successful authentication after multiple random attempts in one minute are less than 1 in 2^77. |

The probabilities for multiple random authentication attempts for the AKEP2 and Needham-Schroeder-Lowe protocols are derived from an upper bound on the data transfer speed between the module's authentication engine and the platform (19.2kB/sec), combined with a lower bound on the amount of data that must be transferred in an authentication attempt (32 octets for AKEP2 and 64 octets for NLS.) If an authentication attempt fails, the protocol must be restarted.

The probabilities for successful random authentication for firmware signature verification are based on the length of the SHA-1 hash (ECDSA uses the P-521 curve, so the hash is the limiting factor.) Each signature verification attempt requires more than 10 seconds to complete; therefore the odds of successful authentication as a result of multiple random attempts within one minute are less than 1 in 2^77.

# 3 ACCESS CONTROL POLICY

This section describes which services and CSP accesses are allowed for each of the module's supported roles.

## 3.1 PRIMARY CRYPTOGRAPHIC OFFICER ROLE

The primary cryptographic officer manages one type of wrapping key (the "Master Key") used to protect client data encryption keys ("Cryptainer Keys".) The primary cryptographic officer may load and unload Master Keys, and request that the module generate Master Keys.

Additionally, the primary cryptographic officer may create and delete cluster officer accounts by requesting the "add/remove cluster officer" service. The primary cryptographic officer may directly access RNG output, load authentication key data into the module, and upgrade the module.

Finally, the primary cryptographic officer may perform all services available to the user and to the unauthenticated platform user.

## 3.2 USER ROLE

The user may load and store key context items, request that the module generate key context items, and encrypt/decrypt client data.

A key context is the set of parameters required to encrypt and decrypt client data. It consists of a data encryption key ("Cryptainer Key"), and non-security relevant items (a block identifier, and two R-strings). The module stores a single key context at any time. Therefore the user is allowed to load and store key context items into the module, and then load data into the module, requesting encryption/decryption with the current key context.

The user does not have access to plaintext Cryptainer Keys, nor to R-strings. These are loaded and stored in encrypted form. A wrapping key and an unwrapping key must first be loaded by a cryptographic officer.

## 3.3 CLUSTER OFFICER ROLE

The cluster officer may authenticate to the module and agree on a Cluster Key (this is a shared wrapping key). Authentication and key agreement is combined into a single service.

## 3.4 UPGRADE FIRMWARE ROLE

The Decru identity can perform the Upgrade service. The Upgrade service consists of loading a new firmware package into the SEP, which verifies the ECDSA signature on the package (external load test.) Authentication and firmware replacement are combined into a single service. This service ends in a mandatory reboot. No other cryptographic operations may be performed during the execution of this service.

## 3.5 UNAUTHENTICATED SYSTEM USER ROLE

The unauthenticated system user represents the platform prior to authentication (the untrusted platform). In practice, this role is performed by a software driver as part of its power on configuration of the module. These services do not disclose, modify, substitute CSPs or use Approved security functions.

Services are made available prior to authentication for the following reasons:

- The platform must configure the SEP during the module's power on process in order to communicate with the module (for example, the platform must assign memory addresses to the module's registers, exercise physical interfaces, and set an interrupt policy for the device.)

- It is desirable to have a facility to zeroize key material in both the platform and the embedded module in any state. This is because the Decru platform features a chassis intrusion detector that may issue a tamper alert even when the module is in a low power state (in which no user is authenticated.)

## 3.6 ROLES AND SERVICES

This is a high level description of all services provided by the module. Because the module is controlled by a low level driver interface, most services encapsulate a set of commands.

**Table 5 – Services Authorized for Roles**

| Role | Authorized Services |
|---|---|
| **Primary cryptographic officer:** | Authenticate System User:<br>Performs an AKEP2 protocol with the operator. This service may also be used to re-authenticate the System User in order to transition from user role to primary cryptographic officer role.<br><br>Enable user services:<br>This command enables the Descriptor Interface. |

| Role | Authorized Services |
|---|---|
| **Primary cryptographic officer (cont.):** | Assume user role:<br>This command restricts the operator's privileges to only those available to the user or to the unauthenticated system user. If user services have not been enabled, then the operator is logged out of the unit.<br><br>Enter AKEP2 AKS:<br>The operator enters new authentication key data into the module for use in authenticating the System User. The old AKS values are no longer used.<br><br>Output Identity (secure):<br>The module exports its ECC-521 public key and its ID to the operator through the secure System – SEP channel.<br><br>Add cluster officer:<br>The operator authorizes an entity to assume the cluster officer role by inserting the officer's public key into the module.<br><br>Remove cluster officer:<br>The operator revokes the public key of a cluster officer.<br><br>Generate Identity:<br>This service results in the creation of an ECC-521 public key pair and SEP ID. No data is output.<br><br>Output random value:<br>The module exports PRNG output to the operator through the secure channel.<br><br>Generate Master Key:<br>The module generates a Master Key. No data is output. |

| Role | Authorized Services |
|---|---|
| **Primary cryptographic officer (cont.):** | Enter Master Key:<br>The operator loads a Master Key into the unit through the System User secure channel. The operator specifies whether the Master Key is to be wrapped before entering the channel.<br><br>Output Master Key:<br>The module sends a Master Key to the operator through the secure channel. The operator specifies whether the key is to be wrapped before entering the channel. |
| **User:** | Enter Key Context item:<br>The module loads key context item(s) into the Key Unit. The Cryptainer Key must be wrapped with a Master Key or with a Cluster Key. Seed values must be wrapped with a Cryptainer Key.<br><br>Generate Key Context item:<br>The module generates key context items and loads them into the key unit. The operator specifies the item(s) to be generated.<br><br>Output Key Context item:<br>The module exports the current key context item(s) to the operator. The operator specifies which entries are to be exported, and how the Cryptainer Key should be wrapped. The wrapping key must be loaded into the unit as a result of an appropriate officer command.<br><br>Encrypt data:<br>The module imports plaintext client data from the operator, and encrypts the data with the AES-256 ECB using the currently loaded key context. The module outputs the ciphertext. |

| Role | Authorized Services |
|---|---|
| **User (cont.):** | Decrypt data:<br>The module imports ciphertext client data from the operator, and encrypts the data with the AES-256 ECB using the currently loaded key context. The module outputs the plaintext. |
| **Cluster Officer:** | Authentication and Key Agreement:<br>Successful authentication shall result in an AES-256 Cluster Key. |
| **Upgrade Firmware:** | Upgrade:<br>Loads new firmware into the module. This service includes performing the external load test. |
| **Unauthenticated System User:** | Zeroize:<br>The operator may specify whether all CSPs, or only those in RAM are to be destroyed.<br><br>Perform power on self-tests:<br>Performed automatically as a result of booting the device.<br><br>Show status:<br>This service corresponds to a suite of commands which return the module's status:<br>• amount of PRNG output in the SEP's PRNG-2 buffer<br>• the value of the module's internal state machines<br>• the public key and/or ID of the module<br>• the public keys of authorized Cluster Members<br>• version of the currently loaded encryption and decryption Master Key(s)<br>• PCI status register values |

| Role | Authorized Services |
|---|---|
| **Unauthenticated System User (cont.):** | Reset:<br>Allows the operator to reset a user-specified number of the module's internal states to their initial value.<br><br>Logout all users:<br>zeroizes all CSPs in the module's RAM, and logs out all users<br><br>Configure module:<br>This service is performed every boot, and sets the register address spaces, interrupt policy, latency settings, and other PCI bus configuration parameters, as documented in the *SEP Reference Manual*.<br><br>Access interfaces:<br>The module provides interfaces to the following external components: DDRAM, I2C bus, 4 LEDs, LCD.<br><br>Read/write to flash:<br>The operator may read from any flash address, and may write to allowed flash addresses.<br><br>Read/write to SDRAM:<br>The module provides a battery-backed, physically protected RAM store for the platform's use. The platform may access this store at any time – no cryptographic processing occurs through this interface. |

| Role | Authorized Services |
|---|---|
| **Unauthenticated System User (cont.):** | Fill SEP PRNG-2 FIFO<br>The platform must ensure that the SEP has sufficient PRNG input stored in its FIFO. No data is output and no keys are created as a result of this service.<br><br>Tamper Notification:<br>The platform may issue a tamper alert to the SEP with this service. |

## 3.7 DEFINITION OF CRYPTOGRAPHIC KEYS AND CRITICAL SECURITY PARAMETERS

The following keys, cryptographic key components and other critical security parameters are contained in the module. Each parameter is followed by a description of the key type/length and the how the item is used. Each name is followed by a symbol that identifies the item.

### 3.7.1 PERSISTENT KEY COMPONENTS AND CSPS

Persistent Key Components are those that remain in the module across a reboot. Items listed in the table below accompanied by an asterisk (*) are public values and they are not protected from disclosure, but are protected from unauthorized modification and substitution. All other items listed in the table below are CSPs that are protected against unauthorized disclosure, modification, and substitution.

**Table 6: Persistent Data**

| Name/Symbol | Use | Format/length |
|---|---|---|
| SEP Confidentiality Key<br>SEP.CK | Wrap/unwrap<br>Master Key | AES-256 bits<br>HMAC-SHA-1<br>(32 octet key) |
| SEP Authentication Key<br>SEP.AK | | |
| Decru Initial Derivation Key<br>Decru.IDK | Derive<br>Sys.IAKS | 80 octets (ANSI KDF) |
| *Decru Public Key<br>Decru.PubKey | ECDSA | ECC-521 |
| *SEP Public Key<br>SEP.PubKey | ECIES | ECC-521 |
| | NSL | |
| SEP Private Key<br>SEP.PrivKey | ECIES | ECC-521 |
| | NSL | |
| *SEP ID<br>SEP.ID | NSL | 16 octets |
| System User<br>Initial Authentication Key Set | AKEP2 | derivation key \|\|<br>authentication key |
| Sys.IAKS | KCDF | (see Sys.DK,<br>Sys.AK) |
| System User<br>Key Derivation Key<br>Sys.DK | AKEP2 | 60 octets (ANSI KDF) |
| System User Authentication<br>Sys.AK | HMAC-SHA-1 | 20 octets (HMAC-SHA-1) |
| | AKEP2 | |
| *Cluster Member Public Key<br>Cl.PubKey | ECIES | ECC-521 |
| | NSL | |
| *Cluster Member ID<br>Cl.ID | NSL | ECC-521 |

### 3.7.2 RUNTIME KEYS

Runtime keys are those stored in the module in RAM, but not destroyed after use.

**Table 7: Runtime Components**

| Name<br>Symbol | Use | Format / length |
|---|---|---|
| Cluster Key<br>Cl.WK | Wrap/Unwrap Clustered Cryptainer Key | AES-256 |
| System User Session Confidentiality Key<br>Sys.SCK | send/receive through Sys.channel | AES-256 |
| System User Session Authentication Key<br>Sys.SAK | send/receive through Sys.channel | HMAC-SHA-1 (20 octet key) |
| Cryptainer Key<br>CK | Encrypt data (default) | AES-256 |
| R-Cryptainer Key<br>RCK | Wrap/Unwrap R1R2 strings | AES-256 |
| Master Key<br>MK | Wrap/Unwrap Cryptainer Key | AES-256 |

### 3.7.3 TEMPORARY KEYS

The following key data exists only briefly during the unit, and is consumed/destroyed after use.

**Table 8: Temporary Components**

| Name<br>Symbol | Use | Format / length |
|---|---|---|
| *Ephemeral public key<br>*ECIES.EPubKey* | ECIES | ECC-521 |
| Ephemeral Private key<br>*ECIES.EPrivKey* | ECIES | ECC-521 |
| ECIES shared secret<br>*ECIES.Z* | ECIES | KDF2 |
| ECIES Encryption Key<br>*ECIES.ECK* | ECIES | AES-256 |
| ECIES Authentication Key<br>*ECIES.EAK* | ECIES | HMAC-SHA-256 |
| *NSL nonces<br>*NSL.N1*<br>*NSL.N2* | NSL | 32 octet nonce |
| *AKEP2 nonces<br>*AKEP2.N1*<br>*AKEP2.N2* | AKEP2 | 32 octet nonce |
| AKEP2 Ephemeral Derivation Key Component<br>AKEP2.EDC | KDF | 20 octets |
| Private values used in group exponentiation<br>AKEP2.DH1, AKEP2.DH2 | DH | 128 octets |

# 3.8 ALGORITHMS AND PROTOCOLS

## 3.8.1 FIPS APPROVED CRYPTO ALGORITHM ENGINES

All approved crypto algorithm engines are assumed to implement critical security functions. Engines are named according to the approved algorithm, followed by

the abbreviations **(SW)** for software implementation or **(HW)** for hardware implementation depending on where the engine is implemented within the cryptographic boundary. In some instances, there are multiple implementations of each algorithm.

- SHA-1(SW) - Cert #192
- SHA-1(HW) - Cert #190, #191
- SHA-256(SW) – Cert #223
- HMAC-SHA-1(SW) – HMAC vendor affirmed as conforming to FIPS 198 using SHA-1 Cert #192
- HMAC-SHA-256(SW) – HMAC vendor affirmed as conforming to FIPS 198 using SHA-256 Cert #223. Only used within an authentication technique and a commercially available key establishment protocol; not used for protection of data.
- AES-256(SW) – Cert #98 CBC mode
- ECDSA(SW) – Vendor affirmed conforming to FIPS 186-2 change notice 1, Appendix 6
- FIPS 186-2 change notice 1, Appendix 3.1 RNG(SW) – Vendor affirmed
- AES-256(HW) – Cert #97, #99 ECB mode

### 3.8.2 NON-APPROVED CRYPTO ENGINES

The following non-approved crypto algorithms and protocols are in the SEP:

- TRNG(SW) hardware random number generator. Only used as a seeding mechanism for the Approved RNG and never used to generate keys directly.

- NSL protocol, conformant to [LOWE]. The protocol makes use of ECIES encryption. ECIES encryption is conformant to [SECG], and incorporates security guidance from [SHOUP]. ECIES encryption makes use of the previously identified AES-256(SW), HMAC-SHA-256(SW) engines, as well as an [X9.63] conformant publicly known non-reversible function based on the previously identified SHA-256(SW) engine.

  This is a commercially available key establishment protocol as allowed under FIPS PUB 140-2 Annex D.

- AKEP2 protocol, conformant to [BR]. The protocol makes use of a SHA-1 based publicly known non-reversible function conformant to [X9.63] and Diffie-Hellman conformant to [X9.42]. This is a commercially available key establishment protocol as allowed under FIPS PUB 140-2 Annex D.

# 3.9 AUTHENTICATION AND KEY ESTABLISHMENT

## 3.9.1 NSL (AUTHENTICATE CLUSTER OFFICER)

The module employs a Needham-Schroeder-Lowe (NSL) protocol to authenticate cluster officers and for key agreement. The system is designed so that one SEP may agree on keys with another SEP. Therefore the module may be in the initiator or receiver mode (this is a mutual authentication protocol.)

## 3.9.2 AKEP2 (AUTHENTICATE SYSTEM USER)

The module uses an implementation of the AKEP2 mutual authentication protocol (that facilitates commercially available key establishment and authentication). The module is always in the initiator mode.

In the module's implementation, the nonces are expanded to contain Diffie-Hellman parameters, a pseudo random function is instantiated as a KDF that incorporates the resulting DH shared secret. Therefore the PRF must be derived before the session keys may be established, and this derivation step is performed by deriving the DH shared secret according to ANSI guidelines [X9.42].

### 3.9.3 KEY TRANSPORT

The following terms shall be used in the remainder of this document:

import is a generic term that refers to transferring data from the System User (platform software) to the SEP.

export is a generic term that refers to transferring data from the SEP to the System User.

load shall refer to sending data to the SEP from the System User while in user role.

store shall refer to transferring data from the SEP to the System User while in user role.

## 3.10 ACCESS RIGHTS WITHIN SERVICES

The following table defines the relationship between access to CSPs and the different module services, using the definitions of modes of access described previously.

If a service listed previously does not appear in the following table, then no CSPs are accessed in that service.

Because the services are presented as groups of commands, with possible parameter inputs, the set of CSP accesses may depend on the service inputs. For instance, when accessing the “zeroize” service, the operator may select which type of CSP to zeroize. In this case, the set of all possible CSP access operations are listed for each service.

When a complex CSP Access operation is performed (e.g. ECIES,) the Access Mode is listed rather than all corresponding CSP read and write operations. In this case, the “Access Type” field is listed as X, for execute. In order to determine all corresponding CSP read/write operations, refer to the definition of each operation.

**Table 9 – Access Rights within Services**

| Service | Cryptographic Keys and CSPs Access Operation | Type(s) of Access |
|---|---|---|
| Zeroize | *Destroy all CSPs listed in sections 3.7.* | Write |
| Reset | Destroy *all CSPs listed in* Table 7 | Write |
| Generate Identity | Generate ECC-521 key pair | Write |
| Authenticate System User | AKEP2 | X |
| | Derive *Sys.IAKS* | X |
| Authentication and Key Agreement | NSL | X |
| Logout all users | Destroy *all CSPs listed in* Table 7 | Write |

| Service | Cryptographic Keys and CSPs Access Operation | Type(s) of Access |
|---|---|---|
| Enter Key Context item | Load wrapped Cryptainer Key | X |
| | Load wrapped R-Cryptainer Key | X |
| | Load wrapped Clustered Cryptainer Key | X |
| | Load wrapped R1R2 strings | X |
| Generate Key Context Item | Generate Cryptainer Key | Write |
| | Generate R1R2 strings | Write |
| Output Key Context Item | Wrap Cryptainer Key | X |
| | Wrap R1R2 strings | X |
| | Wrap Clustered Cryptainer Key | X |
| Encrypt data | Encrypt data | X |
| Decrypt data | Decrypt data | X |
| Fill SEP FIFO | Generate TRNG output | Execute/write |
| | Generate PRNG output | Execute/write |
| Assume user role | Destroy *Sys.SCK* | Write |
| | Destroy *Sys.SAK* | Write |
| Generate Master Key | Generate Master Key | Write |

| Service | Cryptographic Keys and CSPs Access Operation | Type(s) of Access |
|---|---|---|
| Tamper Notification | Destroy Sys.SCK | Write |
| | Destroy Sys.SAK | Write |
| | Destroy *MK* | Write |
| | Destroy *CK* | Write |
| | Destroy *RCK* | Write |
| | Destroy *Cl.WK* | Write |
| Enter Master Key | Receive Master Key | X |
| | Receive Wrapped Master Key | X |
| | Unwrap Master Key | X |
| Output Master Key | Send Master Key | X |
| | Wrap Master Key | X |
| | Send wrapped Master Key | X |
| Add cluster officer | Receive ECC-521 Public Key | X |
| | Receive ID | X |
| Remove cluster officer | Destroy ECC-521 Public Key | Write |
| | Destroy cluster officer ID | Write |
| Output identity (secure) | Send SEP ID | X |
| | Send SEP ECC 521 Public Key | X |

| Service | Cryptographic Keys and CSPs Access Operation | Type(s) of Access |
|---|---|---|
| Output random value | Send PRNG output | X |
| Enter AKEP2 AKS | Receive Sys.AKS | X |
| Upgrade | Decru's Public Key | Read |
| | Compute signature | Write |
| | Compute hash | Write |
| | ECDSA | X |
| Show Status | SEP ID | Read |
| | Cluster Officer Public Key | Read |
| | SEP ECC-521 Public Key | Read |

## 3.11 SECURITY RULES

This section describes the security rules that the module must enforce. The rules are structured according to FIPS PUB 140-2; the security rules enforce the module's conformance to each of the FIPS PUB requirements.

The cryptographic module design corresponds to the following security rules:

1. The cryptographic module shall only support a FIPS mode of operation. The cryptographic module returns its version number through a status command to indicate the approved mode.

2. The SEP shall provide for five roles: unauthenticated system user, user, upgrade firmware, cluster officer, and primary cryptographic officer. For purposes of the standard, the last two roles are considered crypto officer roles.

3. The SEP shall support up to 31 operators that may each assume the cluster officer role.

4. The SEP shall provide for identity-based authentication. The module shall also provide unauthenticated services that do not disclose, modify, or substitute CSPs or use Approved security functions.

5. The module shall track successful authentication by means of an internal state machine. This state machine controls which services may be performed by the module. The state machine is reset on power off, or as a result of a logout command.

6. The module's error states shall consist of soft and hard errors. On encountering soft errors, the module shall note the error and automatically exit the error state after rejecting the data that has been input or is being processed. On encountering a hard error, the module shall disable interfaces used for cryptographic processing, disable the relevant cryptographic engine, issue an error, and discard any data that has been processed during the error state.

7. The module shall not support a bypass or maintenance state.

8. The module shall generate CSPs from the output of a FIPS approved PRNG. This PRNG shall be continuously reseeded by a TRNG, which shall undergo the continuous RNG self-test.

9. All CSPs and public keys within the module shall be protected by the physical security of the device. No CSP shall be output from or entered into the module in plaintext.

10. Only the System User may enter keys into the module.

11. No key may be entered into the module until the System User has authenticated.

12. Master Keys and Key Context items shall be stored only in RAM and shall be zeroized as a result of the logout all users command.

13. The SEP shall provide a service to zeroize all CSPs by the unauthenticated platform user at any time.

14. On power on, the SEP shall perform the following self-tests

    a. AES-256 KATs
    b. SHA-1 KATs
    c. SHA-256 KAT
    d. KDF KAT
    e. HMAC-SHA-1 KAT
    f. KDF2 KAT
    g. HMAC-SHA-256 KAT
    h. Diffie-Hellman KAT
    i. ECCDSA KAT (signature verification only)
    j. PRNG KAT
    k. RNG statistical tests (for both the TRNG and the PRNG)
    l. ECIES test
    m. Software/firmware integrity tests
    n. Test to see if a tamper notice has been issued from the platform
    o. Verification of integrity of stored keys
    p. EDCs attached to the SEP.CK || SEP.AK as well as to Decru.PubKey shall be verified during the module's power on self-tests.

15. Subsequent to power on, both the TRNG and the PRNG shall perform the continuous RNG test. Should a test fail, the module shall notify the operator of the error by writing to a status register, and the module shall discard the error (the module may send additional notifications to the operator.)

16. The module shall include an upgrade service, whereby new firmware is loaded into the SEP. In this case, the module shall perform an external load test, computing the SHA-1 hash of the entire upgrade package. The result of the hash shall be compared with the ECDSA signed hash provided by the Decru. If the signature is verified, and if signed hash matches the hash computed by the module, then the module shall boot from the new firmware on subsequent power on. The cryptographic module shall not support the loading or execution of non-trusted code. Loading of any code that is properly signed with ECDSA, but not validated will invalidate the FIPS 140-2 validation. As such, the area 6 operational environment requirements are not applicable.

# 4 PHYSICAL SECURITY

The SEP is protected with a hard, opaque tamper evident epoxy coating. With high probability, removal of this coating will destroy the underlying circuitry.

**Table 10 – Inspection/Testing of Physical Security Mechanisms**

| Physical Security Mechanisms | Recommended Frequency of Inspection/Test | Inspection/Test Guidance Details |
|---|---|---|
| Hard opaque tamper evident epoxy. | Upon installation of device within the host system. | Thoroughly inspect the cryptographic module for any signs of tamper including scratches, gouges and other suspicious marks on the potting. The device is to be physically destroyed in the event that tamper evidence is noted. |

# 5 MITIGATION OF OTHER ATTACKS

No claims are made about the mitigation of other attacks outside of the scope of FIPS 140-2.

**Table 11 – Mitigation of other attacks**

| Other Attacks | Mitigation Mechanism | Specific Limitations |
|---|---|---|
| N/A | N/A | N/A |

# 6 DEFINITIONS AND ACRONYMS

| TERM | DEFINITION |
|---|---|
| **AKEP2** | Authentication and Key Exchange Protocol, (version) 2. |
| **client** | Refers to an initiator device in a network storage protocol such as NFS. |
| **client data** | Data belonging to a client (that may be stored on a remote server). |
| **cluster** | A set of DataForts (platform + SEP) that share data encryption keys in order to provide failover. |
| **Cryptainer Key** | The key used to encrypt client data |
| **DataFort** | The DataFort platform, with the DCC inserted. The DataFort platform is marketed as a hardware encryption and authenticated device. The SEP is the primary cryptographic service provider for the DataFort. |
| **DataFort platform** | Also called "platform". A computer (CPU, memory, motherboard) in which the DCC may be inserted. There is a unique platform per SEP. The platform serves as the primary SEP operator ("System User"). |

| TERM | DEFINITION |
|---|---|
| **DCC** | Decru Crypto Card -- a PCI card that houses the SEP together with non cryptographic components such as DDRAM, a battery charger, etc. |
| **DDR** | Dual Data Rate memory |
| **Descriptor** | A command to either encrypt/decrypt data, or to load/store a data encryption key. |
| **Descriptor Interface** | Refers to the process of creating descriptors in the descriptor table and writing client data to appropriate memory regions in the platform. This interface is the primary cryptographic interface of the module, and is used to encrypt/decrypt client data. |
| **DMA** | Direct Memory Access. The SEP inputs/outputs client data via direct memory access to either platform memory or on-card DDRAM. |
| **ECIES** | Elliptic Curve Integrated Encryption System, a method to encrypt data using a point on an elliptic curve. |
| **FLASH** | Persistent memory (used to store SEP firmware) |
| **I2C** | A protocol that allows multiple devices to be connected along a single bus. |
| **Key Context** | Data needed to encrypt client data: A Cryptainer Key and information to identify data blocks uniquely. |
| **LCD** | Liquid Crystal Display |

| TERM | DEFINITION |
|---|---|
| **LED** | Light Emitting Diode |
| **Master Key** | A key that encrypts Cryptainer Keys |
| **NAS** | Network Attached Storage. There is a NAS configuration of the SEP that is optimized for the data transfer patterns in NAS file server protocols. |
| **NSL** | Needham-Schroeder-Lowe authentication protocol. |
| **Platform** | The section of the DataFort appliance that is outside of the SEP. |
| **R-strings** | A value used to label file data blocks. |
| **SAN** | Storage Attached Network. A SAN configuration of the SEP includes interfaces for oncard DDRAM (for improved throughput.) |
| **SDRAM** | Persistent memory within the SEP that is made available as a convenience to the platform. |
| **SEP** | Storage Encryption Processor. The SEP is a multi-chip embedded module whose primary purpose is hardware encryption of data. |
| **System User** | Refers to software controlling the DataFort platform. |

# 7 REFERENCES FOR NON-APPROVED CRYPTOGRAPHIC ALGORITHMS/PROTOCOLS

[BR] M. Bellare and P. Rogaway. *Entity Authentication and Key Distribution*. *Advances in Cryptology - CRYPTO 93*, Lecture Notes in Computer Science Vol. 773, D. Stinson, ed., Springer-Verlag, 1994. Available at http://www.cs.ucsd.edu/users/mihir/papers/key-distribution.html

[LOWE] G. Lowe. *Breaking and fixing the Needham-Schroeder public-key protocol using FDR*. In Proc. 2nd International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS), volume 1055 of Lecture Notes in Computer Science, pages 141-166. Springer, 1996.

[SECG] Standards for Efficient Cryptography Group. *SEC1: Elliptic Curve Cryptography* (version 1.0). Available at http://www.secg.org/secg_docs.htm

[SHOUP] V. Shoup. *A Proposal of an ISO Standard for Public Key Encryption* (version 2.1). December 20, 2001. Available at http://www.shoup.net

[X9.42] ANSI X9.42-2003. *Public Key Cryptography for the Financial Services Industry: Agreement of Symmetric Keys Using Discrete Logarithm Cryptography*. 2003. Section 7.5.1.

[X9.63] ANSI X9.63-2001. *Public Key Cryptography for the Financial Services Industry: Key Agreement and Key Transport Using Elliptic Curve Cryptography*. 2001. Section 5.6.3.