# Decru DataFort ™

## SAN SEP v2.0

HW PN/Rev: 60-000191/A, 60-000337/A

FW PN: dcch2_4_2_10_secure

SW PN: 27.8

# *Security Policy*

August 27, 2007

# Changes:

| Revision | Change Description |
|----------|-------------------|
| r4-6 | Section 3.3.8 reworded rule 15p |
| r4-7 | Sec 3.3, Table 3.3 Recover Recovery Policy Key - added RO.SSEK |
| r4-7 | Sec 3.3, Table 3.3 Secret Share Recover Policy Key – added RO.SSEK and removed SEP.PK |
| r4-7 | Sec 3.3, Table 3.3 OTP – added RO.SSAK |
| r4-7 | Sec 3.8 rule 12b replaced System user with operator |
| r4-8 | Sec 3.3 Enter CLU.AKS -- deleted SEP.PK |
| r4-8 | Sec 3.3 Enter Data Domain Key  -- deleted SEP.PK |
| r4-8 | Sec 3.3 Enter Master Key -- changed SEP.SKS to Sys.SKS |
| r4-8 | Sec 3.3 Enter Sys.AKS -- changed SEP.SKS to Sys.SKS |
| r4-8 | Sec 3.3 Output CLU.AKS -- deleted SEP.PK |
| r4-8 | Sec 3.3 Output Master key -- changed SEP.ModDK to SEP.IK and SEP.SKS to Sys.SKS |
| r4-8 | Sec 3.3 Receive CLU.AKS -- Added SEP.PK |
| r4-8 | Sec 3.3 Zeroize: updated to specify whether all CSPs, or only those in RAM are to be destroyed. |
| r4-8 | Sec 3.7changed strength of ECCDH from 260 bits to 256 bits |
| r4-8 | Sec 3.8 Capitalization of roles |
| r4-9 | Sec 1.2.1 updated 'SEP as certified' to 'SEP as validated' |
| r4-9 | Sec 3.8 rule 16. Added – In which case, cryptographic operations stop during the error state. |
| r4-9 | Sec 3.3 updated Enter Recovery Key Set to include SEP.ModDK |
| r4-9 | Sec 3.3 updated Tamper Notification to include SEP.BK, SEP.IK, SEP.PK, Sys.AKS. |
| r4-9 | Reviewed all of the questions. Non-typo fixes consisted of rewording the references to the sys.SKS channel, to clarify the CSP accesses and provide a more uniform description. |

# Table of Contents

# Index of Tables

# 1    INTRODUCTION

The Decru DataFort™ SAN SEP v2.0  is a multi-chip embedded module that is the main cryptographic service provider for Decru DataFort storage encryption product.

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 support the security functional requirements of the Decru DataFort. These are summarized below:

- 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.

- The SEP physically protects plaintext cryptographic keys and CSPs with FIPS 140-2 level 3 physical security requirements.

- The DataFort platform contains a chassis intrusion detector that can force  SEP zeroization (c.f. "Tamper Notification" service).

- Authentication and role enforcement, allowing the platform ("System User") to escalate or de-escalate privileges depending on the amount of key material provided to it. In particular, key access rights are restricted, depending on the role that the platform is in.

## 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 and PCI-X standards. 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 and Figure 2 depict the primary and secondary sides of the DCC including the SEP cryptographic module. The SEP is the potted portion of the card included within the (superimposed) dotted red border. Note that the DCC contains other components that are outside of the cryptographic boundary.
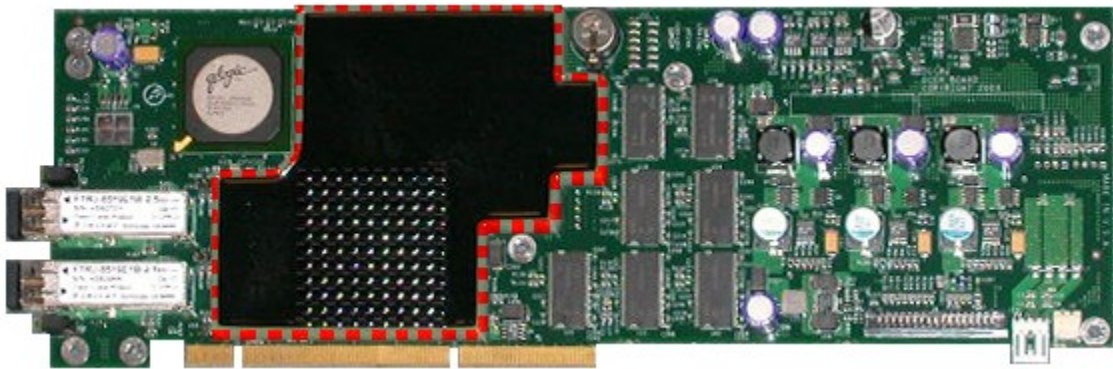


Figure 1: DCC (including SEP module) primary side
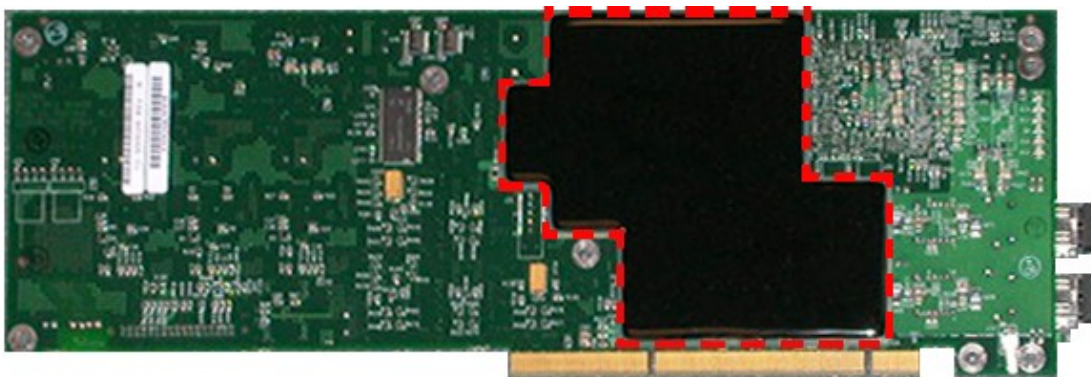


Figure 2: DCC (including SEP module) secondary side

## 1.2.1 Crypto Module Configuration

The SEP as validated has the following configuration:

| | |
|---|---|
| HW PN/Rev: | 60-000191/A |
| | 60-000337/A |
| FW PN: | dcch2_4_2_10_secure |
| SW PN: | 27.8 |

The module is labeled with its configuration in two ways:

- The HW PN/Rev is visible from a label attached to the module. Moreover, the HW PN/Rev, FW PN, and SW PN are reported by the module on every boot via the status output interface.

- DataFort platform administrators can issue the "`sys ver`" CLI command to obtain all part numbers, or they can obtain this information from the WebUI as well. Please consult the *Decru DataFort Administration Guide* for more information about administrative interfaces.

## 1.2.2 Ports and Interfaces

The following table maps the module's physical ports to the FIPS interface classification.

Table 1.1: Interface Classification

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

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 1.2: Security Level

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

| Security Requirements Section | Level |
|---|---|
| Physical Security | 3 |
| 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 five authenticated operator roles, listed in Table 2.1.

Table 2.1: Roles and Required Identification and Authentication

| Identity | Role | Type of Authentication | Authentication Data |
|---|---|---|---|
| System User Identity | User | Identity-based operator authentication | Message authentication key (20 octet HMAC-SHA-1) used in an AKEP2 protocol |
|  | Primary Cryptographic Officer |  |  |
|  | Recovery Officer | Identity-based operator authentication | Message authentication key (20 octet HMAC-SHA-1) used in an AKEP2 protocol AND secret share key RO.SSAK(32 octet AES-256) used in an OTP (one time password) protocol. |
| Cluster Officer Identity | Cluster Officer | Identity-based operator authentication | Message authentication key (20 octet HMAC-SHA-1) used in an AKEP2 protocol |
| Decru Identity | Upgrade Firmware | Identity-based operator authentication | ECC-521 ECDSA signature verification key, AND the ECDSA signature (using SHA-512) of a firmware upgrade package |

## 2.1 The System User Identity

The System User corresponds to the entity controlling the DataFort platform in which the DCC (with embedded module) is inserted. The System User assumes the roles of User, Primary Cryptographic Officer, and Recovery Officer.  There may be only a single System User per module with each role corresponding to an escalation of privileges.

When the module first boots, no operators are authenticated. The System User must authenticate to the module via an AKEP2 protocol. Successful authentication places this operator into the Primary Cryptographic Officer role and also establishes a secure channel to the SEP. The System User is considered to be in the Primary Crypto Officer role as long as the secure channel is open (i.e. The session key material is in the module). While the secure channel is open, the System User may elect to escalate privileges to the Recovery Officer Role by completing the OTP (One Time Password) service, or the System User may elect to reduce privileges by assuming the User role (accomplished by closing the secure channel).

## 2.2 Cluster Officer Identity

The SEP supports up to 31 Cluster Officers. Each Cluster Officer is identified by a unique ID and an HMAC-SHA-1 authentication key. Cluster officer authentication is through the commercially available AKEP2 protocol.

Cluster Officers are remote operators of the module, and correspond to remote SEPs with which the module may share bulk encryption keys. The module supports only a single concurrent Cluster Officer login. Successful authentication of a second Cluster Officer automatically logs out the previous Cluster Officer. When a Cluster Officer is logged into the module, the module is in a state of two concurrent operators (System User and Cluster Officer). The module differentiates between the remote and local operator by providing distinct services to each, issued through separate interfaces. Cluster Officer services are provided through a secure channel with the Cluster Officer encrypted with the platform key .

Cluster Officers may not log into the module unless the platform is in authenticated state (i.e. the System User is in one of the three authenticated roles.) If the System user logs out of the module, any remote Cluster Officer is also logged out by the module.

## 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 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.

## 2.4 Strength of Authentication

Table 2.2 summarizes the strengths of the authentication and authorization protocols.

Table 2.2: Strengths of Authentication Mechanisms

| Mechanism | Strength of Mechanism |
|---|---|
| AKEP2 | The odds of successful random authentication are 1 in $2^{80}$. The odds of successful authentication after multiple random attempts in one minute are less than one in $2^{66}$. |
| AKEP2 + OTP | The odds of successful random authorization attempts are less than 1 in $2^{256}$. The odds of successful authorization after multiple random attempts in one minute are less than one in $2^{242}$. |
| Firmware signature verification (SHA-512/ECDSA) | 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 1 in $2^{253}$. |

# 3 ACCESS CONTROL POLICY

## 3.1 Roles

Table 3.1: Roles

| Role | Description |
|------|-------------|
| Recovery Officer Role | The main services available to the Recovery Officer role are: <br><br> • request that key data be backed up and exported via a secret sharing process <br><br> • Load (previously backed up) key material into the module <br><br> • Load Cluster Officer authentication key material into the module <br><br> • establish a key transfer channel, or "link" corresponding to a specially typed Domain Key (Link Domain Key). This key may be used to wrap and unwrap shareable Cryptainer Keys, allowing, for example, two SEPs to share a specific Cryptainer Key, without being in a cluster (and consequently sharing all Cryptainer Keys.) |
| Primary Cryptographic Officer Role | The Primary Cryptographic Officer manages one type of wrapping key (the "Domain Keys") used to protect client data encryption keys ("Cryptainer Keys".) The Primary Cryptographic Officer may load and unload Domain Keys, and request that the module generate Master Keys. <br><br> 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, and load authentication key data into the module. <br><br> Finally, the Primary Cryptographic Officer may perform all services available to the user and to the unauthenticated platform user. |

| Role | Description |
|------|-------------|
| User Role | The user may load and store key context items, request that the module generate key context items, and encrypt/decrypt client data.<br><br>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.<br><br>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. |
| 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. |
| 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. |
| Unauthenticated System User Role | The unauthenticated system user represents the platform prior to authentication (the untrusted platform).  In practice, this role is accessed via the DataFort platform as part of its power on configuration of the module. Unauthenticated System User Role services do not disclose, modify, substitute CSPs or use Approved security functions.<br><br>Services are made available prior to authentication for the following reasons:<br><br>• 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.)<br><br>• 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.) |

## 3.2 Services and role access rights

Table 3.2 gives a high level description of all services provided by the module and lists the roles allowed to invoke each service.  Because the module is controlled by a low level driver interface, most services encapsulate a set of commands. The following abbreviations are used for roles:

U – Unauthenticated System User role

RO – Recovery Officer role

AU – User role

ClO – Cluster Officer role

CrO – Primary Cryptographic Officer role

FU – Firmware Upgrade role

Table 3.2: Services Authorized for Roles

| U | AU | CrO | RO | ClO | FU | Service Name | Service Description |
|---|----|-----|----|----|----|--------------|---------------------|
| X | X | X | | | | Authenticate System User | 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. |
| X | X | X | | | | Configure module | This service is performed every boot, and sets the register address spaces, interrupt policy, latency settings, and other PCI bus configuration parameters |
| X | X | X | | | | Logout all users | zeroizes all CSPs in the module's RAM, and logs out all users |
| X | X | X | | | | Perform power on self-tests | Performed automatically as a result of booting the device. |
| X | X | X | | | | Read/write to flash | The operator may read from any flash address, and may write to allowed flash addresses. |
| X | X | X | | | | Read/write to SDRAM | 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. |
| X | X | X | | | | Reset | Allows the operator to reset a user-specified number of the module's internal states to their initial value. |

| U | AU | CrO | RO | ClO | FU | Service Name | Service Description |
|---|----|----|----|-----|----|--------------|--------------------|
| X | X | X |  |  |  | Show status | This service corresponds to a suite of commands which return the module's status:<br><br>• the value of the module's internal state machines<br><br>• PCI status register values |
| X | X | X |  |  |  | Tamper Notification | The operator may issue a tamper alert to the SEP with this service. The alert results in a zeroization and logout of all operators. |
| X | X | X |  |  |  | Zeroize | The operator may specify whether all CSPs, or only those in RAM are to be destroyed. |
|  | X | X | X |  |  | Authenticate Cluster Officer | Performs an AKEP2 protocol with the Cluster Identity. The service may only be invoked by one of the three authenticated System User roles. Note that Cluster Officer authentication is revoked if the System User transitions to the unauthenticated role. |
|  | X | X |  |  |  | Decrypt data | The module imports ciphertext client data from the operator, and decrypts the data with the AES-256 ECB using the currently loaded key context. The module outputs the plaintext. |
|  | X | X |  |  |  | Disable user services | Suspends user data encryption and decryption. |
|  | X | X |  |  |  | Encrypt data | 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. |
|  | X | X |  |  |  | Enter Data Domain Key | The operator loads an encrypted Data Domain Key into the module. The Data Domain Key may only be used for encrypting Key Context items. |
|  | X | X |  |  |  | Enter Key Context item | The module loads key context item(s) into the Key Unit. The Cryptainer Key must be wrapped with a Data Domain Key. |
|  | X | X |  |  |  | Enter Link Key | The operator loads an encrypted link key into the module, allowing the module to output exportable Cryptainer Keys by wrapping them with the link key. |

| U | AU | CrO | RO | ClO | FU | Service Name | Service Description |
|---|----|-----|----|-----|----|--------------|---------------------|
|   | X  | X   |    |     |    | Fill SEP FIFO | 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. |
|   | X  | X   |    |     |    | Generate Key Context item | The operator specifies the item(s) to be generated. |
|   | X  | X   |    |     |    | Output Data Domain Key | The module output an encrypted Data Domain Key that is used to encrypt Cryptainer Keys or Link Keys. |
|   | X  | X   |    |     |    | Output Key Context item | The module exports the current key (encrypted) context item(s) to the operator. The operator specifies which entries are to be exported. The wrapping key must be loaded into the unit as a result of an "Enter Data Domain Key" or "Receive Data Domain Key from Cluster Officer" service invocation. |
|   | X  | X   |    |     |    | Output Link Key | The module outputs an encrypted key used to wrap exportable Cryptainer Keys. |
|   |    | X   |    |     |    | Assume user role | 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. |
|   |    | X   |    |     |    | Enable user services | This command enables user data encryption and decryption. |
|   |    | X   |    |     |    | Establish Platform Key | The module establishes a platform key via the ECCDH protocol. |
|   |    | X   |    |     |    | Enter CLU.AKS | The operator loads the encrypted authentication data for Cluster Officers into the module, effectively signifying that the module may attempt to authenticate to the corresponding Cluster Officer user. |
|   |    | X   |    |     |    | Enter Master Key | The operator loads the Master Key encrypted with the Ignition Key into the module, through the secure channel. |
|   |    | X   |    |     |    | Enter Module Domain Key | The operator loads the Module Domain Key encrypted with the Recovery Policy Key into the unit. The Module Domain Key is used to encrypt module secrets, such as secret share keys. |
|   |    | X   |    |     |    | Enter Recovery Policy Key | The operator loads the Recovery Policy Key encrypted with the Master Key into the module. |

| U | AU | CrO | RO | ClO | FU | Service Name | Service Description |
|---|----|-----|----|-----|----|--------------|---------------------|
|   |    | X   |    |     |    | Enter Sys.AKS | 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. |
|   |    | X   |    |     |    | Establish Ignition Key | The module establishes an ignition key and blade key via the ECCDH protocol. |
|   |    | X   |    |     |    | Generate Domain Key | The module generates a Domain Key (consisting of an encryption and HMAC component) together with an ID and type information. |
|   |    | X   |    |     |    | Generate Initial CLU.AKS | The module generates initial authentication key data for the AKEP2 protocol to be run with a Cluster Officer. |
|   |    | X   |    |     |    | Generate Master Key | The module generates a Master Key. If the Ignition Key and blade key are not already in the module, they are generated as a result of this command. No data is output. |
|   |    | X   |    |     |    | Generate Recovery Policy Key | The module generates a Recovery Policy Key, with the policy specified by the operator. The policies determine which interfaces may be used to import the key during a secret recovery operation. |
|   |    | X   |    |     |    | Output CLU.AKS | The module outputs an encrypted Cluster Officer authentication Key Set. |
|   |    | X   |    |     |    | Output Master Key | The module sends a Master Key to the operator through the secure channel. The key is wrapped with the SEP's ignition key. |
|   |    | X   |    |     |    | Output Module Domain Key | The module outputs an encrypted Module Domain Key that is used to encrypt module secrets. |
|   |    | X   |    |     |    | Output random value | The module exports PRNG output to the System User through the established secure channel encrypted with the Platform Key. |
|   |    | X   |    |     |    | Output Recovery Policy Key | The module outputs an encrypted Recovery Policy Key to the operator. |
|   |    | X   |    |     |    | Receive CLU.AKS | The module receives a CLU.AKS replacement from a Cluster Officer, through the channel with the officer encrypted with the Platform Key. |
|   |    | X   |    |     |    | Secret Share Recovery Policy Key | The module subdivides an internally stored Recovery Policy Key into secret shares with a specified recovery threshold, encrypts the shares, and exports them to the operator. |

| U | AU | CrO | RO | ClO | FU | Service Name | Service Description |
|---|----|----|----|----|----|--------------|---------------------|
|   |    | X  |    |     |    | Send CLU.AKS | The module sends the CLU.AKS key to a Cluster Officer by encrypting it with the Platform Key. |
|   |    | X  |    |     |    | Set Crypto Shred Mode | The operator specifies the type of CSPs that are zeroized as part of the tamper notification service. |
|   |    | X  |    |     |    | Set Quorum Requirements | The operator specifies the requirements for a quorum (secret recovery threshold). |
|   |    |    | X  |     |    | Enter Recovery Key Set | The operator loads the encrypted Recovery Key set into the module. Two interfaces are available for this key load: the recovery key set may be encrypted with the SEP.PK and loaded through a channel based on Sys.SKS, or the recovery key set may be encrypted and authenticated with the SEP.ModDK. |
|   |    |    | X  |     |    | Assume Crypto Officer role | Transitions the System User from Recovery Officer role to Primary Cryptographic Officer role. |
|   |    |    | X  |     |    | Establish Link Key | The module performs an ECCDH key agreement protocol with data supplied by the operator to establish a link key. |
|   |    |    | X  |     |    | OTP | The module attempts to authenticate the operator (in Primary Cryptographic Officer role) with a One Time Password protocol. Successful completion of the OTP protocol places the operator in Recovery Officer role. |
|   |    |    | X  |     |    | Recover Recovery Policy Key | The operator loads secret shares containing the Recovery Policy Key into the SEP. The shares are encrypted with secret share key RO.SSEK. The module reconstructs the Recovery Policy Key. |
|   |    |    |    | X   |    | Receive Data Domain Key from Cluster Officer | The module imports the Data Domain Key with a Cluster Officer, encrypted with the Platform Key. Data Domain Key is used to encrypt client data, not module secrets. |
|   |    |    |    | X   |    | Receive Module Domain Key from Cluster Officer | The module imports the Module Domain Key encrypted with Platform Key from a Cluster Officer. |
|   |    |    |    | X   |    | Send Data Domain Key to Cluster Officer | The module sends the Data Domain Key encrypted with Platform Key to a Cluster Officer. |
|   |    |    |    | X   |    | Send Module Domain Key to Cluster Officer | The module sends the Module Domain Key encrypted with the Platform Key to a Cluster Officer. |

| U | AU | CrO | RO | ClO | FU | Service Name | Service Description |
|---|----|----|----|-----|----|--------------|---------------------|
|   |    |    |    |     | X  | Upgrade | Loads new firmware into the module. This service includes performing the external load test. |

## 3.3  Keys and CSP Access Rights

Table 3.3 defines the relationship between services and CSP accesses. The following access mode abbreviations are used:

**R**EAD – the item is read by the service

**W**RITE – the item is written or updated by the service (including zeroization)

**E**XECUTE – the item is used as part of a cryptographic function by the service

Table 3.3: Access Rights within Services

| Service | Description | Keys and CSPs | Access Type(s) |
|---------|-------------|---------------|----------------|
| Assume Crypto Officer role | Clear the OTP nonce | OTP.N | W |
| Assume User role | Logout the System User by clearing the System User session key set. | Sys.SKS | W |
| Authenticate Cluster Officer | Clu.AKS is used for authentication of the Cluster Officer and establishment of the Clu.SKS (session key set). AKEP2.DH1 is an ephemeral parameter to the operation. | Clu.AKS | E |
| | | AKEP2.DH1 | WE |
| | | Clu.SKS | W |
| Authenticate System User | Sys.AKS is used for authentication of the System User and establishment of the Sys.SKS (session key set). AKEP2.DH1 is an ephemeral parameter to the operation. | Sys.AKS | E |
| | | AKEP2.DH1 | WE |
| | | Sys.SKS | W |
| Decrypt data | User data is decrypted with SEP.CK | SEP.CK | E |
| Encrypt data | User data is encrypted with SEP.CK | SEP.CK | E |
| Establish Platform Key | SEP.PK is established based on SEP.ECCDHSecret | SEP.ECCDHSecret | E |
| | | SEP.PK | W |
| Enter CLU.AKS | Clu.AKS is verified and decrypted with SEP.ModDK | SEP.ModDK | E |
| | | Clu.AKS | W |

| Service | Description | Keys and CSPs | Access Type(s) |
|---|---|---|---|
| Enter Data Domain Key | SEP.DataDK is verified and decrypted with SEP.RPK | SEP.RPK | E |
|  |  | SEP.DataDK | W |
| Enter Key Context item | SEP.CK is verified and decrypted with SEP.DataDK | SEP.DataDK | E |
|  |  | SEP.CK | W |
| Enter Link Key | SEP.LK is verified and decrypted with SEP.ModDK | SEP.ModDK | E |
|  |  | SEP.LK | W |
| Enter Master Key | SEP.MK is verified and decrypted with SEP.IK. This service is executed through the Sys.SKS channel. | SEP.IK | E |
|  |  | Sys.SKS | E |
|  |  | SEP.MK | W |
| Enter Module Domain Key | SEP.ModDK is verified and decrypted with SEP.RPK | SEP.RPK | E |
|  |  | SEP.ModDK | W |
| Enter Recovery Policy Key | SEP.RPK is verified and decrypted with SEP.MK | SEP.MK | E |
|  |  | SEP.RPK | W |
| Enter Sys.AKS | SEP.AKS is decrypted with SEP.PK. This service is executed through the Sys.SKS channel. | SEP.PK | E |
|  |  | Sys.SKS | E |
|  |  | Sys.AKS | W |
| Establish Ignition Key | SEP.IK is established based on SEP.ECCDHSecret | SEP.ECCDHSecret | E |
|  |  | SEP.BK | E |
|  |  | SEP.IK | W |
| Establish Link Key | Link key is established based on SEP.ECCDHSecret | SEP.ECCDHSecret | E |
|  |  | SEP.BK | E |
|  |  | SEP.LK | W |
| Generate Data Domain Key | Generation from PRNG output | SEP.DataDK | W |
| Generate Initial CLU.AKS | Generation from PRNG output | Clu.AKS | W |
| Generate Key Context item | Generation from PRNG output | SEP.CK | W |
| Generate Master Key | Generation from PRNG output | SEP.MK | W |
|  |  | SEP.IK | W |
| Generate Module Domain Key | Generation from PRNG output | SEP.ModDK | W |
| Generate Recovery Policy Key | Generation from PRNG output | SEP.RPK | W |
| Logout all users | Zeroize all runtime CSPs (see Table 3.4) | All run-time CSPs | W |

| Service | Description | Keys and CSPs | Access Type(s) |
|---|---|---|---|
| OTP | Run One Time Password protocol. The RO.SSAK key is used to encrypt the challenge. The response data to this service is sent through the Sys.SKS channel. | OTP.N | RWE |
| | | RO.SSAK | E |
| | | Sys.SKS | E |
| Output CLU.AKS | Clu.AKS is encrypted and authenticated with SEP.ModDK | Clu.AKS | R |
| | | SEP.ModDK | E |
| Output Data Domain Key | SEP.DataDK is encrypted and authenticated with SEP.RPK | SEP.RPK | E |
| | | SEP.DataDK | R |
| Output Key Context item | SEP.CK is encrypted and authenticated with SEP.DataDK | SEP.DataDK | E |
| | | SEP.CK | R |
| Output Link Key | SEP.LK is encrypted and authenticated with SEP.ModDK | SEP.ModDK | E |
| | | SEP.LK | R |
| Output Master Key | SEP.MK is encrypted and authenticated with SEP.IK, and output through the Sys.SKS channel. | SEP.IK | E |
| | | Sys.SKS | E |
| | | SEP.MK | R |
| Output Module Domain Key | SEP.ModDK is encrypted and signed with SEP.RPK | SEP.RPK | E |
| | | SEP.ModDK | R |
| Output random value | PRNG output is output via the Sys.SKS channel. | Sys.SKS | E |
| Output Recovery Policy Key | SEP.RPK is encrypted and signed with SEP.MK | SEP.MK | E |
| | | SEP.RPK | R |
| Receive CLU.AKS | Clu.AKS is decrypted with the SEP.PK. The service is executed through the Clu.SKS channel. | Clu.SKS | E |
| | | SEP.PK | E |
| | | Clu.AKS | W |
| Receive Data Domain Key from Cluster Officer | SEP.DataDK is decrypted with the SEP.PK. The service is executed through the Clu.SKS channel. | Clu.SKS | E |
| | | SEP.PK | E |
| | | SEP.DataDK | W |
| Receive Module Domain Key from Cluster Officer | SEP.ModDK is decrypted with the SEP.PK. The service is executed through the Clu.SKS channel. | Clu.SKS | E |
| | | SEP.PK | E |
| | | SEP.ModDK | W |
| Recover Recovery Policy Key | SEP.RPK is entered into the module encrypted with RO.SSEK. | RO.SSEK | E |
| | | SEP.RPK | W |
| Secret Share Recovery Policy Key | SEP.RPK is encrypted with RO.SSEK | RO.SSEK | E |
| | | SEP.RPK | R |

| Service | Description | Keys and CSPs | Access Type(s) |
|---|---|---|---|
| Send CLU.AKS | Clu.AKS is encrypted with the SEP.PK, and exported. This service is executed through the Clu.SKS channel. | Clu.SKS | E |
| | | SEP.PK | E |
| | | Clu.AKS | R |
| Enter Recovery Key Set | Loads, RO.SSEK and RO.SSAK, and decrypts with the SEP.PK, executed through the Sys.SKS channel, or loads RO.SSEK and RO.SSAK decrypted and verified with SEP.ModDK. | SEP.PK | R |
| | | Sys.SKS | E |
| | | SEP.ModDK | E |
| | | RO.SSEK | W |
| | | RO.SSAK | W |
| Send Data Domain Key to Cluster Officer | SEP.DataDK is encrypted with SEP.PK, and output through the Clu.SKS channel. | Clu.SKS | E |
| | | SEP.PK | E |
| | | SEP.DataDK | R |
| Send Module Domain Key to Cluster Officer | SEP.ModDK is encrypted with SEP.PK, and output through the Clu.SKS channel. | Clu.SKS | E |
| | | SEP.PK | E |
| | | SEP.ModDK | R |
| Tamper Notification | If crypto shred policy set to 2 or 3, zeroize all runtime CSPs (see Table 3.4). If crypto shred policy set to 1, zeroize all CSPs, except Decru.IDK, SEP.BK, SEP.IK, SEP.PK, Sys.AKS. | * | W |
| Upgrade | Decru signature on upgrade code verified with Decru.pubKey | Decru.pubKey | E |
| Zeroize | The operator may specify whether all CSPs, or only those in RAM are to be destroyed. | * | W |

## 3.4  CSPs

The following keys, cryptographic key components and other critical security parameters are contained in the module. Each CSP is assigned a row in Table 3.4. The following interpretation applies to the persistence column:

> persistent – the key/CSP persists until explicit zeroization

> runtime – the key/CSP persists until explicit zeroization or power-cycle

> ephemeral – the key/CSP is used for a single algorithm instance only (e.g. nonces)

Table 3.4: Cryptographic Keys and CSPs

| CSP Symbol | CSP Name | Lifetime | Type |
|---|---|---|---|
| Decru.IDK | Decru Initial Derivation Key | persistent | Authentication key (20 octets), Key derivation key (60 octets) |
| SEP.PK | SEP Platform Key | runtime | AES-256, HMAC-SHA-256 |
| SEP.IK | SEP Ignition Key | persistent | AES-256, HMAC-SHA-256 |
| SEP.BK | SEP Blade Key | persistent | AES-256, HMAC-SHA-256 |
| Sys.AKS | System User Authentication Key Set | persistent | authentication key (20 octets), Key derivation key (60 bytes) |
| Sys.SKS | System User Session Key Set | runtime | AES-256, HMAC-SHA-1 1 byte channel sequence number |
| Clu.AKS | Cluster User Authentication Key Set | runtime | HMAC-SHA-256 (40 octets), Key derivation key (40 octets) |
| Clu.SKS | Cluster User Session Key Set | runtime | AES-256, HMAC-SHA-256, 1 byte channel sequence number |
| SEP.MK | SEP Master Key | runtime | AES-256 (32 octets) HMAC-SHA-256 (32 octets) |
| SEP.RPK | SEP Recovery Policy Key | runtime | AES-256 (32 octets) HMAC-SHA-256 (32 octets) |
| RO.SSEK | Recovery Secret Share Encryption Key | runtime | AES-256 (32 octets) HMAC-SHA-1 (20 octets) |
| RO.SSAK | Recovery Secret Share Authorization Key | runtime | AES-256 (32 octets) HMAC-SHA-1 (20 octets) |
| SEP.ModDK | SEP Module Domain Key | runtime | AES-256 (32 octets) HMAC-SHA-256 (32 octets) |
| SEP.DataDK | SEP Data Domain Key | runtime | AES-256 (32 octets) HMAC-SHA-512 (32 octets) |
| SEP.CK | SEP Cryptainer Key | runtime | AES-256 (32 octets) HMAC-SHA-512 (32 octets) |
| SEP.LK | SEP Link Key | runtime | AES-256 (32 octets) HMAC-SHA-512 (32 octets) |
| AKEP2.EDC | AKEP2 Ephemeral Derivation Key Component | ephemeral | 20 octet key derivation component |
| AKEP2.DH1 | AKEP2 Diffie-Hellman private key | ephemeral | Diffie-Hellman private value used for group exponentiation (128 octets) |
| SEP.ECCDHSecret | ECC Diffie-Hellman secret key | ephemeral | ECC-521 private key |
| OTP.N | One Time Password (OTP) nonce | ephemeral | AES-256 (32 octets) HMAC-SHA-1 (20 octets) |

## 3.5 Public Keys and public nonces

The following table lists public keys and public nonces.

Table 3.5: Public Keys and Nonces

| CSP Symbol | CSP Name | Lifetime | Type |
|---|---|---|---|
| Decru.PubKey | Decru Public Key | persistent | ECC521 ECDSA verification public key |
| AKEP2.N1 AKEP2.N2 | AKEP2 initiator /receiver nonce | ephemeral | 20 octet nonce (when used with Sys.AKS) 32 octet nonce (when used with Clu.AKS) |

## 3.6 FIPS Approved Crypto Algorithm Engines

All approved crypto algorithm engines are assumed to implement critical security functions.

Table 3.6: Approved cryptographic algorithm implementations

| Algorithm Implementation | References | Certificate(s) |
|---|---|---|
| SHA-1 | FIPS PUB 180-2 | Cert #192 |
| SHA-256 | FIPS PUB 180-2 | Cert #223 |
| SHA-512 | FIPS PUB 180-2 | Cert #511 |
| HMAC-SHA-1 | FIPS PUB 198, uses SHA-1, Cert #192 | Cert #210 |
| HMAC-SHA-256 | FIPS PUB 198, uses SHA-256, Cert #223 | Cert #211 |
| HMAC-SHA-512 | FIPS PUB 198, uses SHA-512, Cert #511 | Cert #212 |
| AES-256-CBC | FIPS PUB 197 | Cert #446 |
| AES-256-ECB | FIPS PUB 197 | Cert #445 |
| ECDSA | FIPS PUB 186-2, change notice 1, Appendix 6. | Cert #35 |
| PRNG | FIPS PUB 186-2, change notice 1, Appendix 3.1. | Cert #232 |

## 3.7 Non-approved Crypto Algorithm Engines

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

Table 3.7: Non-Approved Algorithms

| Algorithm Implementation | Comments |
|---|---|
| TRNG | Hardware random number generator. Only used as a seeding mechanism for the Approved RNG and never used to generate keys directly. |
| AKEP2 protocol | Authentication and Key agreement protocol, conformant to AKEP2. The protocol makes use of a SHA-1 based publicly known non-reversible function conformant to ANSI X9.63 and Diffie-Hellman conformant to ANSI X9.42-2003. The module only relies on this protocol for authentication, it does not use the protocol for key agreement. |
| ECCDH | ECC Diffie-Hellman key agreement protocol conformant to ANSI X9.63. The protocol makes use of the SHA-256 based key derivation function conformant to ANSI X9.63. The ECCDH protocol provides 256 bits of strength. This is a commercially available key agreement protocol as allowed under FIPS PUB 140-2 Annex D. |
| Secret Sharing/ Secret Recovery | A threshold split knowledge scheme as defined in HAC. The scheme is not relied upon to protect CSPs, as the output shares are encrypted prior to export with an approved security function (AES-256). |
| KDF1 | A supporting function for the AKEP2 protocol, conformant to to ANSI X9.63, and based on SHA-1. |
| KDF2 | A supporting function for the ECCDH protocol, conformant to to ANSI X9.63, and based on SHA-256. |

## 3.8 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 six roles: Unauthenticated System User, User, upgrade firmware, Cluster Officer, Primary Cryptographic Officer, and Recovery Officer. For purposes of the standard, the last three roles are considered crypto officer roles.

3.  The SEP shall support multiple operators that may each assume the Cluster Officer role. Only a single operator (the System User) may assume the User, Cryptographic Officer, and Recovery Officer roles.

4.  The SEP shall provide for identity-based authentication. The module shall also provide for zeroization as an unauthenticated service, and other 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. Both the TRNG and the PRNG shall undergo a continuous RNG self-test (see Rule #16).

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. The module shall distinguish between *module* and *user* secrets:

*module secrets* are defined as any of the following:

    a. authentication data associated to one of the supported roles

    b. session keys between the module and a user of the module

    c. secret share keys

    d. platform key

    e. any key in the key hierarchy that encrypts/signs one of the CSPs listed previously in a,b,c.

*user secrets* are defined as any of the following:

    f. Encryption and signature Cryptainer Keys

    g. Those Domain Keys that are used to encrypt and sign Cryptainer Keys.

12. Module secrets may only be loaded into or out of the module when the operator is in one of the following roles:

    a. Primary Cryptographic Officer role

    b. Upgrade Firmware Role

    c. Recovery Officer role

13. The module shall distinguish between the key material it makes available to the System User in Primary Cryptographic Officer role and Recovery Officer role:

    a. In Primary Cryptographic Officer role, the module shall enforce a "black box" CSP access policy in which keys are not accessible to the System User in plaintext, given the key material exchanged by two parties (System User in Primary Cryptographic Officer role, SEP).

    b. In Recovery Officer role, the module shall export keys in a hierarchy, at the top of which are secret share keys, that are accessible to the System User in Recovery Officer role (as this user provided the secret share keys to the module).

14. All CSPs except for

- authentication data between the module and the System User

- The SEP's Ignition Key package

- The upgrade firmware public key, Decru.PubKey

shall be stored only in RAM and shall be zeroized as a result of the logout all users service.

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

 a. AES-256 KATs

 b. SHA-1 KAT

 c. SHA-256 KAT

 d. SHA-512 KAT

 e. KDF1 KAT

 f. HMAC-SHA-1 KAT

 g. KDF2 KAT

 h. HMAC-SHA-256 KAT

 i. HMAC-SHA-512 KAT

 j. Diffie-Hellman KAT

 k. ECCDH KAT

 l. ECDSA (signature verification only), SHA-512 KAT hash verification with Decru.PubKey

 m. PRNG KAT, in which the PRNG is initialized with a fixed seed value and internal state and the output of the PRNG is compared against a stored value.

 n. Software/firmware integrity tests

 o. Test to see if a tamper notice has been issued from the platform

 p. SHA-1 attached to keys stored in the EPROM shall be verified during the module's power on self-tests.

16. Subsequent to power on, both the TRNG and the PRNG shall perform the continuous RNG test. The TRNG shall store and compare 8 octets for the continuous test, and the PRNG shall store and compare 40 octets. Should a test fail, the module shall enter an error state during which no cryptographic operations can be performed, notify the operator of the error by writing to a status register, abandon all buffered RNG bits, and then exit the error state.

17. 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-512 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 the 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.

18. Unless key material is loaded into the module through the secure channel from the System User, the module shall not contain sufficient key material to perform the encrypt data or decrypt data service. In particular, Cryptainer Keys may not be compromised as a result of compromising the physical security of a powered off module.

19. Prior to initialization, the module shall allow only the System User to authenticate using ephemeral and default key material. During this session, the only allowed services are:

   a.  services available to the unauthenticated user

   b.  *Output random value* (in order to reseed the PRNG of the System User)

   c.  *Enter Sys.AKS* (in order to change to a local AKS).

   d.  *Fill SEP FIFO*

Thereafter, the System User must shut down the session and re-authenticate with the new AKS in order to access the full set of services. Prior to the AKS change, Cluster Officers and Recovery Officers may not authenticate to the module.

## 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 4.1: 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 5.1: Mitigation of other attacks

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

## 6    DEFINITIONS AND ACRONYMS

| | |
|---|---|
| *AKEP2* | Authentication and Key Exchange Protocol, (version) 2. The module uses AKEP2 for authentication, it does not use AKEP2 for key agreement. |
| *Authentication Key Set (AKS)* | A key derivation key, together with an authentication key. The authentication key is used in an AKEP2 protocol, and on success, the key derivation key is used to derive a session key set. This module only uses AKEP2 for authentication, it does not use AKEP2 for key agreement. |
| *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 DataFort appliances that share data encryption keys in order to provide failover and load balancing. |
| *Cryptainer Key* | The key package used to encrypt and sign 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"). |
| *DCC* | Decru Crypto Card – a PCI card that houses the SEP together with non cryptographic components such as DDRAM, a battery charger, etc. |
| *Domain Key* | A key package used to encrypt and sign CSPs such as Cryptainer Keys. |

| | |
|---|---|
| *Ignition Key* | A key package used to encrypt and sign the Master Key prior to exporting it to the System User. |
| *Link Key* | A key package that is used to encrypt and sign certain exportable Cryptainer Keys. |
| *Master Key* | A key package used to encrypt and sign Recovery Policy Keys. |
| *Platform* | The section of the DataFort appliance that is outside of the SEP. |
| *quorum* | The minimum number of secret shares required to reconstitute the secret that is being shared. Sometimes referred to as a threshold. |
| *Recovery Officer* | A System User role that is allowed to recover key material, establish link keys, and establish trust between Cluster Officers. |
| *Recovery Policy Key* | A key package used to encrypt and sign Domain Keys. |
| *secret share key* | A key package used to encrypt and sign certain Recovery Policy Keys. |
| *SEP* | Storage Encryption Processor. The SEP is a multi-chip embedded module whose primary purpose is hardware encryption of data. |
| *Session Key Set* | A wrapping key, together with an initial chaining value |
| *System User* | Refers to the host machine i.e. DataFort platform. |
| *Wrapping Key* | A key consisting of an encryption and an HMAC signature component (i.e. two distinct keys) |

# 7    REFERENCES

AKEP2                          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

ANSI 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.

ANSI 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.

FIPS PUB 140-2                 Security Requirements for Cryptographic Modules,  2001 May 25, Change
                               Notice 4, 2002 Dec 03.

FIPS PUB 180-2                 Secure Hash Standard (SHS), 2002 August

FIPS PUB 186-2                 Digital Signature Standard (DSS),  2000 January 27. Change Notice 1, 2001
                               October 5.

FIPS PUB 197                   Advanced Encryption Standard (AES), 2001 November 26

FIPS PUB 198                   The Keyed-Hash Message Authentication Code (HMAC), 2002 March

HAC                            *Handbook of Applied Cryptography.* Alfred J. Menezes, Paul C. Van
                               Oorschot, Scott A. Vanston. CRC Press, August 2001. Section 12.7.2.