TITLE:	Luna® K3 Cryptographic Engine Security Policy Model
ABSTRACT:	This document describes the security policies enforced by the Luna K3 cryptographic engine.
DOCUMENT NUMBER:	CR-1423
ORIGINATOR:	Terry Fletcher
DEPARTMENT:	Engineering
LOCATION OF ISSUE:	Ottawa
DATE ORIGINATED:	21 March 2002
CHANGE LEVEL:	8
CHANGE DATE:	May 31, 2005
SECURITY LEVEL:	
SUPERSESSION DATA:	CR-1423, 7

TABLE OF CONTENTS

1. INTRODUCTION 5
1.1. Purpose 5
1.2. Scope 5
1.3. History of Revision 5
1.4. References 5
1.5. Glossary of Acronyms / Abbreviations 5
2. SECURITY POLICY MODEL INTRODUCTION 6
2.1. Functional Overview 6
2.2. Assets to be Protected 8
2.3. Operating Environment 8
3. SECURITY POLICY MODEL DESCRIPTION 9
3.1. Operational Policy 9
3.1.1. Module Capabilities 10
3.1.2. Partition Capabilities 11
3.2. FIPS-Approved Mode 15
3.3. Description of Operator, Subject and Object 16
3.3.1. Operator 16
3.3.2. Roles 16
3.3.3. Account Data 16
3.3.4. Subject 17
3.3.5. Operator – Subject Binding 17
3.3.6. Object 17
3.3.7. Object Operations 17
3.4. Identification and Authentication 18
3.4.1. Authentication Data Generation and Entry 18
3.4.2. Trusted Path 18
3.4.3. Limits on Login Failures 18
3.4.4. M of N Activation 19
3.5. Access Control 19
3.5.1. Object Re-use 21
3.5.2. Privileged Functions 21
3.6. Cryptographic Material Management 21
3.7. Cryptographic Operations 22
3.8. Information Flow Control 22
3.9. Firmware Security 23
3.10. Physical Security 23
3.11. Fault Tolerance 23
3.12. Backup And Recovery 23
3.13. Mitigation of Other Attacks 24
APPENDIX A. CRYPTOGRAPHIC ALGORITHMS SUPPORT 25
APPENDIX B. SECURITY POLICY CHECKLIST TABLES 27

List of Figures

Figure 2-1 Luna® SA HSM Server System 6
Figure 2-2 K3 Cryptographic Module Inside Luna SA 7
Figure 2-3 Block Diagram of Components Showing Cryptographic Module 8

List of Tables

Table 3-1 Module Capabilities and Policies 12
Table 3-2 Partition Capabilities and Policies 13
Table 3-3 Object Attributes Used in Access Control Policy Enforcement 20
Table B-1 Roles and Required Identification and Authentication 27
Table B-2 Strengths of Authentication Mechanisms 27
Table B-3 Services Authorized for Roles 27
Table B-4 Access Rights within Services 27

1. INTRODUCTION

1.1. Purpose

This document describes the security policies enforced by the Luna®K3 Cryptographic Engine, also known as the K3 cryptographic module or K3. This document applies to hardware versions 2.0, 3.0, and 4.0 and firmware version 4.1.0.

1.2. Scope

The security policies described in this document apply to the K3 cryptographic module only and do not include any security policy that may be enforced by the host appliance.

1.3. History of Revision

Revision	Date	Description
Draft	21 March 2002	Draft for circulation
1	8 November 2002	Changes to Module and Partition capabilities and policies; added Appendix A, list of algorithms
2	26 November 2002	Inclusion of Appendix B to satisfy tabular format requirements of FIPS 140-2, Appendix C.
3	6 February 2003	Changes to address lab comments.
4	June 19, 2003	Security Policy changed to include FIPS Level 2 description.
5	August 26, 2003	Changes made to reflect queries from NIST/CSE.
6	December 12, 2003	Removed remote authentication, activation and auto-activation from scope of validation as directed by NIST/CSE.
7	May 18, 2004	Changes made to reflect queries from NIST/CSE.
8	May 31, 2005	Addition of hardware version 4.0.

1.4. References

Document Number	Author	Title
CR-1400	D. Bailey	Key Partition and License HLD
CR-1402	D. Bailey	Login and Auto-activation HLD
CR-1387	C. Dunn	Viper Software Requirements
CR-1385	C. Dunn	Viper Feature Definition

1.5. Glossary of Acronyms / Abbreviations

Term	Explanation
Chrysalis-ITS	Former name of SafeNet Canada
CSP	Certification Service Provider
FPV	Fixed Policy Vector
HA	High Availability
HSM	Hardware Security Module
NTLA	Network Trust Link Agent
NTLS	Network Trust Link Server
PCI	Peripheral Component Interconnect
PED	PIN Entry Device
TPV	Token or K3 Policy Vector

2. SECURITY POLICY MODEL INTRODUCTION

2.1. Functional Overview

The K3 cryptographic engine is a multi-chip embedded hardware cryptographic module in the form of a PCI card that resides within a customized general-purpose computing appliance. It is contained in its own secure enclosure that provides physical resistance to tampering and zeroization in the event the enclosure is opened. The cryptographic boundary of the module is defined to encompass all components inside the secure enclosure on the PCI card. Figure 2-1 illustrates the use of the K3 cryptographic module and associated peripherals, such as the Luna® PED and the backup token, within the Luna® SA HSM Server system (an example of a host appliance using the K3 cryptographic module). Figure 2-2 shows the K3 cryptographic module as it is installed in the Luna SA HSM Server.

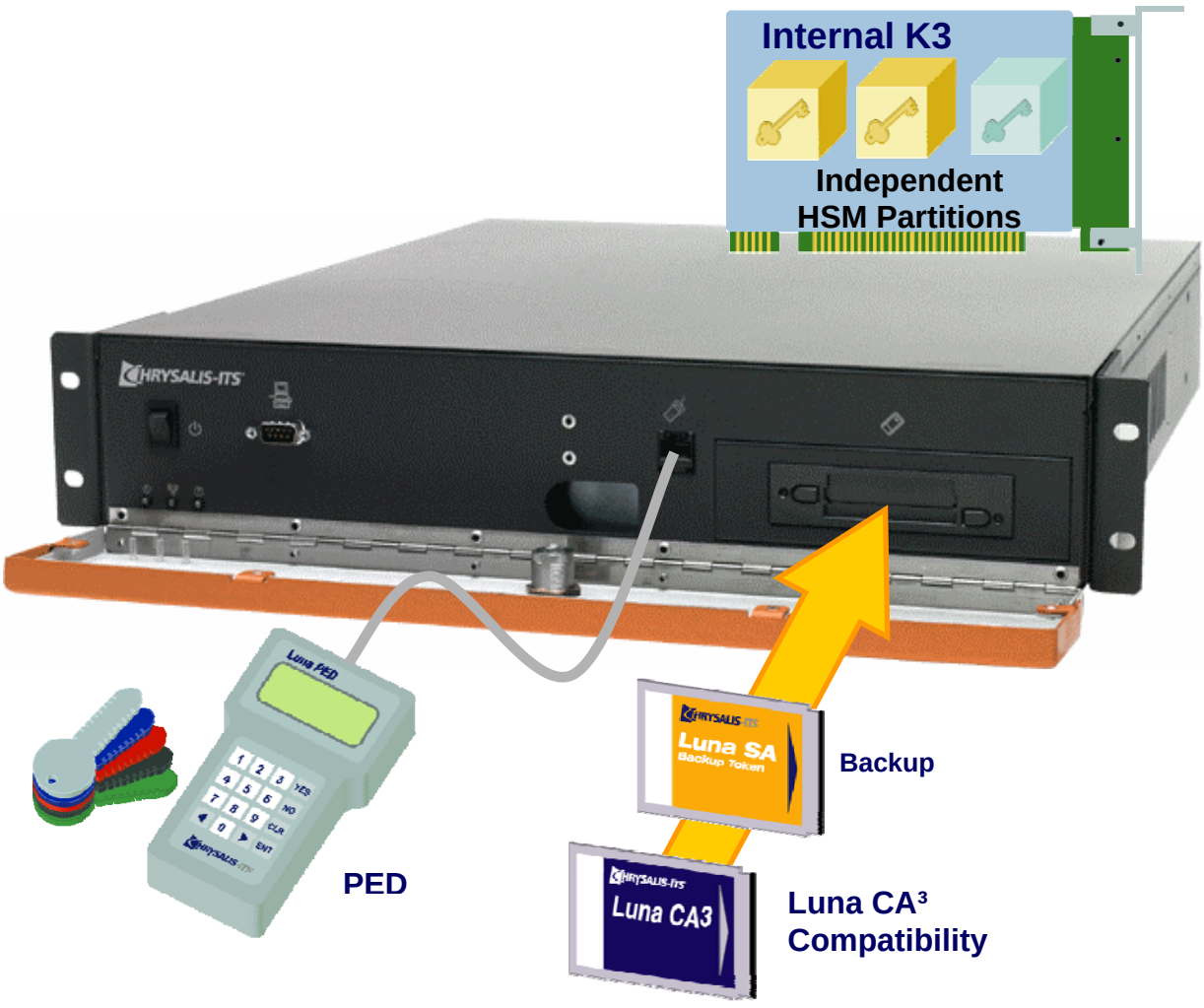

Figure 2-1 Luna® SA HSM Server System

Figure 2-2 K3 Cryptographic Module Inside Luna SA

The module may be explicitly configured to operate in either FIPS Level 2 or FIPS Level 3, or in a non-FIPS mode of operation. Configuration in either FIPS mode enforces the use of FIPS-approved algorithms only. Configuration in FIPS Level 3 mode also enforces the use of trusted path authentication.

The cryptographic module is accessed directly (i.e., electrically) via either the PIN Entry Device (PED) serial interface or via the PCI communications interface. Access for users and user application software is provided indirectly through the host appliance, either locally via a console interface or remotely via a TCP/IP network interface using SSH or SSL V3 -secured connections. The module provides secure key generation and storage for symmetric keys and asymmetric key pairs along with symmetric and asymmetric cryptographic services. It provides the ability to manage multiple user definitions and concurrent authentication states. The client and server software that provide the connections to the module present a logical view of “virtual tokens” or “partitions” to user applications. Each partition must be separately authenticated in order to make it available for use.

In general terms, the module operates in a manner similar to the current SafeNet Canada (formerly known as Chrysalis-ITS) PC Card-based modules and requires similar configuration procedures and operating conditions as the PC Card-based products.

Figure 2-3 depicts the overall system components at the client and appliance (server) sides and shows the portions covered by the validation – the cryptographic module plus the specially modified PC Card Reader that provides the PED interface in the grey shaded boxes.

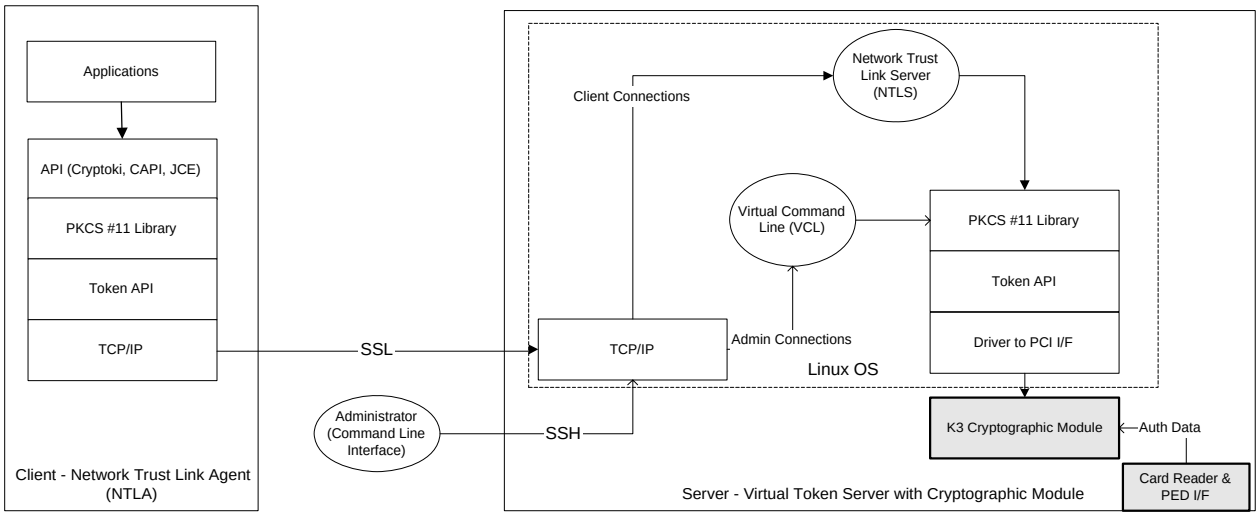

Figure 2-3 Block Diagram of Components Showing Cryptographic Module

2.2. Assets to be Protected

The module is designed to protect the following assets:

1. User-generated private keys,
2. User-generated secret keys,
3. Cryptographic services, and
4. Module security data.

2.3. Operating Environment

The module is assumed to operate as part of a network-attached security appliance that resides on an internal TCP/IP network with no direct Internet connection. It is assumed that the appliance includes an internal host computer that runs an application service that provides a console or secure shell (SSH) interface for use by locally connected or remote administrators and an application service that provides communications services to allow access to the module's cryptographic services by user applications running on host computers connected to the internal network.

The security appliance will include the following:

1. A Pentium III (or better) class general purpose CPU and motherboard (or compatible).
2. A hard disk drive.
3. An Ethernet Network Interface Card.
4. Customized PC Card reader with secure port connection.
5. Suitable operating system with minimal services installed.
6. Tamper detection switches on the appliance enclosure.
7. Cryptographic module.

8. PIN Entry Device.

It is assumed that the internal host computer of the security appliance is running a properly hardened version of a suitable operating system (such as Linux) and that only necessary application services are installed. It is also assumed that only known secure versions of the application services are permitted to run on the internal host computer of the appliance.

It is assumed that the appliance enclosure provides control over physical access to the module, including tamper detection and response mechanisms for attempts to open the appliance enclosure. Response at this level means de-activating the cryptographic module to render the module functions and sensitive data inaccessible to an intruder and preventing login to the module until the tamper condition is no longer present.

It is assumed that trained and trustworthy administrators are responsible for the initial configuration and ongoing maintenance of the appliance and the cryptographic module.

It is assumed that access to the cryptographic module will be controlled, either by accessing the module via a direct local connection or by accessing it via remote connections, controlled by the client and server software providing the connection.

3. SECURITY POLICY MODEL DESCRIPTION

This section provides a narrative description of the security policy enforced by the module. It is intended both to state the security policy enforced by the module and to give the reader an overall understanding of the security behaviour of the module. The detailed functional specification for the module is provided elsewhere.

The security behaviour of the cryptographic module is governed by the following security policies:

- Operational Policy
- Identification and Authentication Policy
- Access Control Policy
- Cryptographic Material Management Policy
- Information Flow Control Policy
- Firmware Security Policy
- Physical Security Policy

These policies complement each other to provide assurance that cryptographic material is securely managed throughout its life cycle and that access to other data and functions provided by the product is properly controlled. Configurable parameters that determine many of the variable aspects of the module's behaviour are specified by the higher level Operational Policy implemented at two levels: the cryptographic module as a whole and the individual partition. This is described in section 3.1.

The Identification and Authentication policy is crucial for security enforcement and it is described in section 3.4. The access control policy is the main security functional policy enforced by the module and is described in section 3.5, which also describes the supporting object re-use policy. Cryptographic Material Management is described in section 3.6. The information flow control policy governs the transfer of information between entities within the scope of policy enforcement and is described in section 3.8. Firmware security, physical security, fault tolerance and backup/recovery are described in sections 3.9 through 3.12.

3.1. Operational Policy

The module employs the concept of the Operational Policy to control the overall behaviour of the module and each of the partitions within. At each level, either the module or the partition is assigned a fixed set of "capabilities" that govern the allowed behaviour of the module or individual partition. The SO establishes the

Operational Policy by enabling/disabling or refining the corresponding policy elements to equate to or to be more restrictive than the pre-assigned capabilities.

The set of configurable policy elements is a proper subset of the corresponding capability set. That is, not all elements of the capability set can be refined. Which of the capability set elements have corresponding policy set elements is pre-determined based on the "personality" of the partition or manufacturing restrictions placed on the module. For example, the module capability setting for "domestic algorithms and key sizes available" does not have a corresponding configurable policy element.

Further, policy set elements can only refine capability set elements to more restrictive values. Even if an element of the policy set exists to refine an element of the capability set, it may not be possible to assign the policy set element to a value other than that held by the capability set element. Specifically, if a capability set element is set to allow, the corresponding policy element may be set to either enable or disable. However, if a capability set element is set to disallow, the corresponding policy element can only be set to disable. Thus, an SO cannot use policy refinement to lift a restriction set in a capability definition.

In comparing the Operational Policy of the K3 cryptographic module to that of the Luna® token-based products, the reader will note similarities between the capability set(s) and the Fixed Policy Vector (FPV) settings of the token-based products, and also between the policy set(s) and the Token Policy Vector (TPV) of the token-based products. Although similarities exist, there are also fundamental differences between the K3 cryptographic module's sets and the token-based FPV and TPV. In particular, there are several FPV settings that do not have corresponding capability set elements. These are elements of the cryptographic module's behaviour that are truly fixed and, therefore, are no longer subject to configuration in the capability sets. The specific elements affected are the following:

- Allow/disallow non-sensitive secret keys – fixed as disallow.
- Allow/disallow non-sensitive private keys – fixed as disallow.
- Allow/disallow secret key creation through the create objects interface – fixed as disallow.
- Allow/disallow private key creation through the create objects interface – fixed as disallow.

3.1.1. Module Capabilities

The following is the set of capabilities supported at the module level:

- Allow/disallow non-FIPS algorithms available.
- Allow/disallow password authentication.
- Allow/disallow trusted path authentication.
- Allow/disallow M of N.
- Allow/disallow cloning.
- Allow/disallow masking.
- Allow/disallow M of N auto-activation.
- Allow/disallow domestic algorithms and key sizes available (Default is Allow).
- Allow/disallow modification of personality licenses (Default is Disallow).
- Allow/disallow modification of capabilities[1].
- Allow/disallow ECC mechanisms (Default is Disallow).

[1] This module capability exists only to permit programming of capabilities at SafeNet Canada in the event that appropriate licensing is not possible. It will always be set to "disallowed" when the module is delivered to the customer.

- Performance level (4 bits).
- Number of failed SO logins allowed before the HSM is zeroized (3).

3.1.2. Partition Capabilities

The following is the set of capabilities supported at the partition level. All capability elements described as "allow/disallow some functionality" are Boolean values where false (or zero) equates to disallow the functionality and true (or one) equates to allow the functionality. The remainder of the elements are integer values of the indicated number of bits.

- Allow/disallow partition reset.
- Allow/disallow activation.
- Allow/disallow automatic activation.
- Allow/disallow HA.
- Allow/disallow multipurpose keys.
- Allow/disallow changing of certain key attributes once a key has been created.
- Allow/disallow operation without RSA blinding.
- Allow/disallow signing operations with non-local keys.
- Allow/disallow raw RSA operations.
- Allow/disallow private key wrapping.
- Allow/disallow private key unwrapping.
- Allow/disallow secret key wrapping.
- Allow/disallow secret key unwrapping.
- Allow/disallow Level 3 operation without a challenge.
- Allow/disallow all functionality in excess of that implemented by backup tokens.
- Allow/disallow incrementing of failed login attempt counter on failed challenge response validation.
- Level of storage space available for key storage (4 bits).
- Minimum/maximum password length (applies only to Level 2 modules and minimum must be >= 7).
- Number of failed Partition User logins allowed before partition is locked out/cleared.

The following capabilities are only configurable if cloning is allowed and enabled at the module level:

- Allow/disallow private key cloning.
- Allow/disallow secret key cloning.

The following capabilities are only configurable if masking is allowed and enabled at the module level:

- Allow/disallow private key masking.
- Allow/disallow secret key masking.

In addition, the masking function can only be used according to the following restrictions:

- If cloning is not allowed or not enabled, masking/unmasking can only be used by the original module within its host appliance.
- If cloning is allowed and enabled, masking/unmasking can be used across multiple modules within the same domain.

Table 3-1 Module Capabilities and Policies

Description	Capability	Policy	Comments
Non-FIPS algorithms available	Allow	Enable	SO can configure the policy to enable or disable the availability of non-FIPS algorithms at the time the HSM is initialized.
		Disable	
	Disallow	Disable	The HSM must operate using FIPS approved algorithms only.
Password authentication	Allow	Enable	SO can configure the policy to enable or disable the use of passwords without trusted path for authentication.
		Disable	
	Disallow	Disable	The HSM must operate using the trusted path and module-generated secrets for authentication.
Trusted path authentication	Allow	Enable	SO can configure the policy to enable or disable the use of the trusted path and module-generated secrets for authentication.
		Disable	
	Disallow	Disable	The HSM must operate using passwords without trusted path for authentication.[2]
M of N	Allow	Enable	SO can configure the policy to enable or disable the use of M of N secret sharing to activate the module. Requires that the policy for "trusted path" authentication be enabled.
		Disable	
	Disallow	Disable	The HSM must operate without M of N secret sharing for activation.
Cloning	Allow	Enable	SO can configure the policy to enable or disable the availability of the cloning function for the HSM as a whole.
		Disable	
	Disallow	Disable	The HSM must operate without cloning.
Masking	Allow	Enable	SO can configure the policy to enable or disable the availability of the masking function for the HSM as a whole.
		Disable	
	Disallow	Disable	The HSM must operate without masking.
M of N auto-activation	Allow	Enable	SO can configure the policy to enable or disable the use of the M of N auto-activation feature.
		Disable	
	Disallow	Disable	The HSM must operate without M of N auto-activation.
Domestic product algorithms and key sizes available	Allow	N/A	This capability is set prior to shipment to the customer. It controls the availability of domestic strength algorithms and key sizes. The default setting is Allow.
	Disallow		
Modification of personality licenses	Allow	N/A	This capability is set prior to shipment to the customer. It controls the ability to modify partition personality licenses in the field. The default setting is Disallow.
	Disallow		
Modification of capabilities	Allow	N/A	This capability is set prior to shipment to the customer. It controls the ability to modify the capability set in the field. This capability will only be set to Allow for SafeNet Canada use; it will always be set to Disallow for a module shipped to a customer.
	Disallow		

[2] One and only one means of authentication ("user password" or "trusted path") must be enabled by the policy. Therefore, either one or both of the authentication capabilities must be allowed and, if one of the capabilities is disallowed or the policy setting disabled, then the policy setting for the other must be enabled.

Description	Capability	Policy	Comments
ECC mechanisms available	Allow	N/A	This capability is set prior to shipment to the customer. It controls the availability of ECC mechanisms. The default setting is Disallow.
	Disallow		
Partition reset	Allow	Enable	SO can configure the policy to enable a partition to be reset if it is locked as a result of exceeding the maximum number of failed login attempts.
		Disable	
	Disallow	Disable	A partition cannot be reset and must be re-created as a result of exceeding the maximum number of failed login attempts.
Network Replication	Allow	Enable	SO can configure the policy to enable the replication of the module's key material over the network to a second module.
		Disable	
	Disallow	Disable	The module cannot be replicated over the network.
Non-backup token functions	Allow	Enable	Applies to G3 token-based modules. SO can enable full functionality for the token or simple backup functionality only.
		Disable	
	Disallow	Disable	Backup functionality only is allowed.

Table 3-2 Partition Capabilities and Policies

Description	Prerequisite	Capability	Policy	Comments
Level 3 operation without a challenge	Trusted path authentication enabled	Allow	Enable	SO can configure the policy to enable Level 3 login using the PED trusted path only, with no challenge - response validation required.
			Disable	
		Disallow	Disable	Challenge - response validation required plus PED trusted path login to access the partition.
Count failed challenge – response validations	Trusted path authentication enabled	Allow	Enable	SO can configure the policy to count failures of the challenge - response validation against the maximum login failures or not.
			Disable	
		Disallow	Disable	Failures of the challenge - response validation are not counted against the maximum login failures.
Activation	Trusted path authentication enabled	Allow	Enable	SO can configure the policy to enable the authentication data provided via the PED trusted path to be cached in the module, allowing all subsequent access to the partition, after the first login, to be done on the basis of challenge - response validation alone.
			Disable	
		Disallow	Disable	PED trusted path authentication is required for every access to the partition.
Auto-activation	Trusted path authentication enabled	Allow	Enable	SO can configure the policy to enable the activation data to be stored on the appliance server in encrypted form, allowing the partition to resume its authentication state after a re-start. This is intended primarily to allow partitions to automatically re-start operation when the appliance returns from a power outage.
			Disable	
		Disallow	Disable	Activation data cannot be externally cached.
High Availability	N/A	Allow	Enable	SO can configure the policy to enable

Description	Prerequisite	Capability	Policy	Comments
			Disable	the use of the High Availability login feature. This allows a partition in one appliance to act as a trusted remote authentication device for one or more partitions in remotely connected appliances.
		Disallow	Disable	High Availability login cannot be used.
Multipurpose keys	N/A	Allow	Enable	SO can configure the policy to enable the use of keys for more than one purpose, e.g., an RSA private key could be used for digital signature and for decryption.
			Disable	
		Disallow	Disable	Keys can only be used for a single purpose.
Change attributes	N/A	Allow	Enable	SO can configure the policy to enable changing key attributes.
			Disable	
		Disallow	Disable	Key attributes cannot be changed.
Operate without RSA blinding	N/A	Allow	Enable	SO can configure the use of blinding mode for RSA operations. Blinding mode is used to defeat timing analysis attacks on RSA digital signature operations, but it also imposes a significant performance penalty on the signature operations.
			Disable	
		Disallow	Disable	Blinding mode is not used for RSA operations.
Signing with non-local keys	N/A	Allow	Enable	SO can configure the ability to sign with externally-generated private keys that have been imported into the partition.
			Disable	
		Disallow	Disable	Externally-generated private keys cannot be used for signature operations.
Raw RSA operations	N/A	Allow	Enable	SO can configure the ability to use raw (no padding) format for RSA operations.
			Disable	
		Disallow	Disable	Raw RSA cannot be used.
Private key wrapping	N/A	Allow	Enable	SO can configure the ability to wrap private keys and export them from the partition.
			Disable	
		Disallow	Disable	Private keys cannot be wrapped and exported from the partition.
Private key unwrapping	N/A	Allow	Enable	SO can configure the ability to unwrap private keys and import them into the partition.
			Disable	
		Disallow	Disable	Private keys cannot be unwrapped and imported into the partition.
Secret key wrapping	N/A	Allow	Enable	SO can configure the ability to wrap secret keys and export them from the partition.
			Disable	
		Disallow	Disable	Secret keys cannot be wrapped and exported from the partition.
Secret key unwrapping	N/A	Allow	Enable	SO can configure the ability to unwrap secret keys and import them into the partition.
			Disable	
		Disallow	Disable	Secret keys cannot be unwrapped and imported into the partition.
Private key cloning	Cloning	Allow	Enable	SO can configure the ability to clone

Description	Prerequisite	Capability	Policy	Comments
	enabled, Trusted path authentication enabled		Disable	private keys from one partition to another in the same domain.
		Disallow	Disable	Private keys cannot be cloned.
Secret key cloning	Cloning enabled, Trusted path authentication enabled	Allow	Enable	SO can configure the ability to clone secret keys from one partition to another in the same domain.
			Disable	
		Disallow	Disable	Secret keys cannot be cloned.
Private key masking	Masking enabled	Allow	Enable	SO can configure the ability to mask private keys for storage outside the partition.
			Disable	
		Disallow	Disable	Private keys cannot be masked for storage outside the partition.
Private key unmasking	Masking enabled	Allow	Enable	SO can configure the ability to unmask private keys and retrieve them into the partition.
			Disable	
		Disallow	Disable	Private keys cannot be unmasked and retrieved into the partition.
Secret key masking	Masking enabled	Allow	Enable	SO can configure the ability to mask secret keys for storage outside the partition.
			Disable	
		Disallow	Disable	Secret keys cannot be masked for storage outside the partition.
Secret key unmasking	Masking enabled	Allow	Enable	SO can configure the ability to unmask secret keys and retrieve them into the partition.
			Disable	
		Disallow	Disable	Secret keys cannot be unmasked and retrieved into the partition.
Storage space	N/A			
Minimum/maximum password length	User password authentication enabled	7/16 characters		SO can configure password length for Level 2 modules, but minimum length must always be >= 7.
Number of failed Partition User logins allowed	N/A	10	Configurable	SO can configure; default is 10.

3.2. FIPS-Approved Mode

The SO controls operation of the module in FIPS-approved mode, as defined by FIPS PUB 140-2, by enabling or disabling the appropriate Module Policy settings (assuming each is allowed at the Module Capability level). To operate in FIPS-approved mode, the following policy settings are required:

- "Non-FIPS Algorithms Available" must be disabled,
- Activation and Auto-activation must be disabled,
- Level 3 operation without a challenge must be enabled.

Additionally,

- For operation at **FIPS Level 3**, "Trusted path authentication" must be enabled (implies that password authentication is disallowed or disabled), or
- For operation at **FIPS Level 2**, "User password authentication" must be enabled (implies that trusted path authentication is disallowed or disabled).

The policy settings for "Trusted path authentication" and "User password authentication" may also be configured in the case where "Non-FIPS Algorithms Available" has been enabled.

If the SO selects policy options (e.g., enables “Non-FIPS Algorithms Available”) that would place the module in a mode of operation that is not approved, a warning is displayed and the SO is prompted to confirm the selection. Any time the SO displays the policy settings, a message is displayed indicating that the module is or is not in an Approved mode.

3.3. Description of Operator, Subject and Object

3.3.1. Operator

An operator is defined as an entity that acts to perform an operation on the module. An operator may be directly mapped to a responsible individual or organization, or it may be mapped to a composite of a responsible individual or organization plus an agent (application program) acting on behalf of the responsible individual or organization.

In the case of a Certification Authority (CA), for example, the organization may empower one individual or a small group of individuals acting together to operate the cryptographic module as part of the company’s service. The operator might be that individual or group, particularly if they are interacting with the module locally. The operator might also be the composite of the individual or group, who might still be present locally to the module (particularly for authentication purposes), plus the Certification Authority application running on a network-attached host computer.

3.3.2. Roles

The K3 cryptographic module supports two authenticated operator roles: Partition User and the Security Officer. It also supports one unauthenticated operator role, the Public User, primarily to permit access to status information and diagnostics before authentication. The SO is a privileged role, which exists only at the module level, whose primary purpose is to initially configure the module for operation and to perform security administration tasks such as partition creation. The Partition User is the normal operational role on the module. For an operator to assume either the Partition User or Security Officer role, the operator must be identified and authenticated. The following conditions must hold in order to assume either role:

- No operator can assume either the Partition User or Security Officer role before identification and authentication;
- No identity can assume both a Partition User and a Security Officer role; and
- When M of N Activation has been enabled, as required by local security policy, no operator can assume either the Partition User or the Security Officer role before the M of N activation has been completed.

3.3.3. Account Data

The module maintains the following Partition User and SO account data:

- Partition ID or SO ID number
- Partition User encrypted or SO encrypted authentication data (checkword)
- Partition User authentication challenge secret
- Partition User locked out flag.

The ability to manipulate the account data is restricted to the SO and the Partition User roles. The specific restrictions are as described below:

1. Only the Security Officer role can create (initialize) and delete the following security attributes:
    - Partition ID;
    - Checkword
2. If Partition reset is allowed and enabled, the SO role only can modify the following security attribute:

- Locked out flag for Partition User

3. Only the Partition User role can modify the following security attribute:
    - Checkword for Partition User
4. Only the Security Officer role can change the default value, query, modify and delete the following security attribute:
    - Checkword for Security Officer

3.3.4. Subject

For purposes of this security policy, the subject is defined to be a module session. The session provides a logical means of mapping between applications connecting to the module and the processing of commands within the module. Each session is tracked by Session ID, the Partition ID and the Access ID, which is a unique ID associated with the application's connection. It is possible have multiple open sessions with the module associated with the same Access ID/Partition ID combination. It is also possible for the module to have sessions opened for more than one Partition ID or have multiple Access ID's with sessions opened on the module. Applications running on remote host systems that require data and cryptographic services from the module must first connect via the communications service within the appliance, which will establish the unique Access ID for the connection and then allow the application to open a session with one of the partitions within the module. A local application (e.g., command line administration interface) will open a session directly with the appropriate partition within the module without invoking the communications service.

3.3.5. Operator – Subject Binding

An operator must access a partition through a session. A session is opened with a partition in an unauthenticated state and the operator must be authenticated before any access to cryptographic functions and Private objects within the partition can be granted. Once the operator is successfully identified and authenticated, the session state becomes authenticated and is bound to the Partition User represented by the Partition ID. Any other sessions opened with the same Access ID/Partition ID combination will share the same authentication state and be bound to the same Partition User.

3.3.6. Object

An object is defined to be any formatted data held in volatile or non-volatile memory on behalf of an operator. For the purposes of this security policy, the objects of primary concern are private (asymmetric) keys and secret (symmetric) keys.

3.3.7. Object Operations

The Partition User role is the only one permitted to perform object operations.

New objects can be made in several ways. The following list identifies operations that produce new objects:

- Create,
- Copy,
- Generate,
- Unwrapping,
- Derive.

Existing objects can be modified and deleted. The values of a subset of attributes can be changed through a modification operation. Objects can be deleted through a destruction operation. Constant operations do not cause creation, modification or deletion of an object. These constant operations include:

- Query an object's size;

- Query the size of an attribute;
- Query the value of an attribute;
- Use the value of an attribute in a cryptographic operation;
- Search for objects based on matching attributes;
- Wrapping an object;
- Masking and unmasking an object.

Secret keys and private keys are always maintained as Sensitive objects and, therefore, they are permanently stored with the key value encrypted to protect its confidentiality. Key objects held in volatile memory do not have their key values encrypted, but they are subject to active zeroization in the event of a module reset or in response to a tamper event. Operators are not given direct access to key values for any purpose.

3.4. Identification and Authentication

3.4.1. Authentication Data Generation and Entry

The module requires that Partition Users and the SO be authenticated by proving knowledge of a secret shared by the operator and the module. The FIPS mode (either level 2 or level 3) is determined when the HSM is initialized: a module that is to support level 2 mode must be initialized using a password to define the SO PIN; a module that is to support level 3 mode must be initialized using the PED to define the SO PIN.

For a module operating in FIPS Level 2 mode, the SO must enable the "User password authentication" (implies that the trusted path authentication is disallowed or disabled). The SO defines a user password when a partition is created. The minimum length of the password must always be equal to or greater than 7 characters, and up to 16 characters.

For a module operating in FIPS Level 3 mode, the module generates the authentication secret as a 48-byte random value and, optionally, an authentication challenge secret. The authentication secret(s) are provided to the operator via a physically separate trusted path, described in sub-section 3.4.2, and must be entered by the operator via the trusted path during the login process.

3.4.2. Trusted Path

In FIPS Level 3 mode, authentication must be performed using a trusted path. The authentication data is a 48-byte value that is randomly generated by the module and stored on an authentication device (serial memory device) via the physical trusted path. The data on the authentication device must then be entered into the module via the trusted path as part of each login process.

3.4.3. Limits on Login Failures

The module also implements a maximum login attempts policy. The policy differs for an SO authentication data search and a Partition User authentication data search.

In the case of an SO authentication data search:

- If three (3) consecutive SO logon attempts fail, the module is zeroized.

In the case of a Partition User authentication data search, one of two responses will occur, depending on the partition policy:

1. If “Partition reset” is Allowed and Enabled, then – If “n” (“n” is set by the SO at the time the HSM is initialized) consecutive operator logon attempts fail, the module flags the event in the Partition User’s account data, locks the Partition User and clears the volatile memory space. The SO must unlock the partition and the operator must reset the partition authentication data in order for the Partition User to resume operation.
2. If “Partition reset” is not Allowed or not Enabled, then – If “n” consecutive Partition User logon attempts via the physical trusted path fail, the module will erase the partition. The SO must delete and re-create the partition. Any objects stored in the partition, including private and secret keys, are permanently erased.

3.4.4. M of N Activation

If M of N activation is required by the Module Policy, “M” pieces out of a total of “N” pieces of a split authentication secret must be entered via the trusted path in order to activate the module for operation. The M of N secret and the splits are generated by the module.

3.5. Access Control

The Access Control Policy is the main security function policy enforced by the module. It governs the rights of a subject to perform privileged functions and to access objects stored in the module. It covers the following object operations:

- Create
- Read (Query Attribute Value)
- Copy
- Modify
- Destroy
- Generate
- Derive
- Wrap
- Unwrap
- Use
- Clone
- Mask

A subject’s access to objects stored in the module is mediated on the basis of the following subject and object attributes:

- Subject attributes:
  - Session ID
  - Access ID and Partition ID associated with session
  - Session authentication state (binding to authenticated Partition identity)
- Object attributes:
  - **Owner.** A Private object is owned by the Partition User associated with the subject that produces it. Ownership is enforced via internal key management.
  - **Private.** If True, the object is Private. If False, the object is Public.

- **Sensitive.** If True, object is Sensitive. If False, object is Non-Sensitive.
- **Extractable[3].** If True, object may be extracted. If False, object may not be extracted.
- **Modifiable.** If True, object may be modified. If False, object may not be modified.

Objects are labelled with a number corresponding to their partition and are only accessible by a subject associated with the owning Partition ID. Only generic data and certificate objects can be non-sensitive. Sensitive objects are encrypted using the partition's secret key to prevent their values from ever being exposed to external entities. Key objects are always created as Sensitive objects and can only be used for cryptographic operations by a logged in Partition User. Key objects that are marked as extractable may be exported from the module using the Wrap operation if allowed and enabled in the partition's policy set. Table 3-3 summarizes the object attributes used in Access Control Policy enforcement.

Table 3-3 Object Attributes Used in Access Control Policy Enforcement

Attribute	Values	Impact
PRIVATE	TRUE – Object is private to (owned by) the operator identified as the Access Owner when the object is created.	Object is only accessible to subjects (sessions) bound to the operator identity that owns the object.
	FALSE – Object is not private to one operator identity.	Object is accessible to all subjects associated with the partition in which the object is stored.
SENSITIVE	TRUE – Attribute values representing plaintext key material are not permitted to exist (value encrypted).	Key material is stored in encrypted form.
	FALSE – Attribute values representing plaintext data are permitted to exist.	Plaintext data is stored with the object and is accessible to all subjects otherwise permitted access to the object.
MODIFIABLE	TRUE – The object's attribute values may be modified.	The object is "writeable" and its attribute values can be changed during a copy or set attribute operation.
	FALSE – The object's values may not be modified.	The object can only be read and only duplicate copies can be made.
EXTRACTABLE	TRUE – Key material stored with the object may be extracted from the K3 using the Wrap operation.	The ability to extract a key permits sharing with other cryptomodules and archiving of key material.
	FALSE – Key material stored with the object may not be extracted from the K3.	Keys must never leave the module's control.

The module does not allow any granularity of access other than owner or non-owner (i.e., a Private object cannot be accessible by two Partition Users and restricted to other Partition Users). Ownership of a Private object gives the owner access to the object through the allowed operations but does not allow the owner to assign a subset of rights to other operators. Allowed operations are those permitted by the Module and Partition Capability and Policy settings.

The policy is summarized by the following statements:

- A subject may perform an allowed operation on an object if the object is in the partition with which

[3]Extract means to remove the key from the control of the module. This is typically done using the Wrap operation, but the Mask operation is also considered to perform an extraction when cloning is enabled for the container.

the subject is associated and one of the following two conditions holds:

1. The object is a “Public” object, i.e., the PRIVATE attribute is FALSE, or
2. The subject is bound to the Partition User that owns the object.

- Allowed operations are those permitted by the object attribute definitions within the constraints of the Module and Partition Capability and Policy settings.

3.5.1. Object Re-use

The access control policy is supported by an object re-use policy. The object re-use policy requires that the resources allocated to an object be cleared of their information content before they are re-allocated to a different object.

3.5.2. Privileged Functions

The module shall restrict the performance of the following functions to the SO role only:

- Module initialization
- Partition creation and deletion
- Configuring the module and partition policies
- Module zeroization
- Firmware update

3.6. Cryptographic Material Management

Cryptographic material (key) management functions protect the confidentiality of key material throughout its life-cycle. The FIPS PUB 140-2 approved key management functions provided by the module are the following:

(1) Pseudo random number generation in accordance with ANSI X9.31, Appendix A.

(2) Cryptographic key generation in accordance with the following indicated standards:

a. RSA 1024, 2048 and 4096 bits key pairs in accordance with FIPS PUB 186-2.

b. 3DES 112, 168 bits (FIPS PUB 46-3, ANSI X9.52).

c. AES 128, 192, 256 bits (FIPS PUB 197).

d. DSA 1024 bits key pairs in accordance with FIPS PUB 186-2.

(3) Secure key storage and key access following the PKCS #11 standard.

(4) Destruction of cryptographic keys is performed in one of three ways as described below in accordance with the PKCS #11 and FIPS PUB 140-2 standards:

a. An object on the K3 that is destroyed using the PKCS #11 function C_DestroyObject is marked invalid and remains encrypted with the Partition User's key or the K3’s general secret key until such time as its memory locations (flash or RAM) are re-allocated for additional data on the K3, at which time they are purged and zeroized before re-allocation.

b. Objects on the K3 that are destroyed as a result of authentication failure are zeroized (all flash blocks in the Partition User’s memory turned to 1's). If it is an SO authentication failure all flash blocks used for key and data storage on the K3 are zeroized.

c. Objects on the K3 that are destroyed through C_InitToken (the SO-accessible command to initialize the K3 available through the API) are zeroized, along with the rest of the flash memory being used by the SO and Partition Users.

Keys are always stored as secret key or private key objects with the Sensitive attribute set. The key value is, therefore, stored in encrypted form using the owning Partition User's secret key. Access to keys is never provided directly to a calling application. A handle to a particular key is returned that can be used by the application in subsequent calls to perform cryptographic operations.

Private key and secret key objects may be imported into the module using the Unwrap, Unmask (if cloning is enabled) or Derive operation under the control of the Access Control Policy. Any externally-set attributes of keys imported in this way are ignored by the module and their attributes are set by the module to values required by the Access Control Policy.

3.7. Cryptographic Operations

Because of its generic nature, the module firmware supports a wide range of cryptographic algorithms and mechanisms. The cryptographic functions and algorithms that are relevant to the FIPS 140-2 validation are the following:

(1) Symmetric encryption/decryption (key wrap/unwrap) 3DES 168 bits in accordance with PKCS #11.

(2) Symmetric encryption/decryption 3DES 112, 168 bits (FIPS PUB 46-3, ANSI X 9.52).

(3) Symmetric encryption/decryption AES 128, 192, 256 bits (FIPS PUB 197).

(4) Signature generation/verification RSA 1024, 2048, 4096 (PKCS #1, FIPS PUB 186-2) with SHA-1 (FIPS PUB 180-1, ANSI X9.30 Part 2), DSA 1024 bits (FIPS PUB 186-2) with SHA-1.

(5) Hash generation SHA-1 (FIPS PUB 180-1)

3.8. Information Flow Control

By default, information is not permitted to flow:

1. From one partition to a different partition within the same module, or
2. From a partition on one module to a partition on a different module, or
3. From a partition within a module to any external entity.

Information stored within the module is permitted to flow from one partition to a different partition within the same module or from a partition on one module to a partition on a different module only if one of the following conditions holds:

1. The information is contained within a non-sensitive data object or objects, OR
2. The Module Capability Allow Cloning is set and the corresponding Module Policy element is Enabled, AND
3. At least one of the following Partition Policies is allowed and enabled:
   - Clone private keys.
   - Clone secret keys.
   - Mask private keys (implies that masking is allowed and enabled at the module level).
   - Mask secret keys (implies that masking is allowed and enabled at the module level).

Information stored within the module is permitted to flow from a partition to an external entity only if one of the following conditions holds:

1. The information is contained within a non-sensitive data object or objects, OR
2. The Module Capability Allow Cloning is not set or the corresponding Module Policy element is not Enabled, AND
   - The Partition Policy to wrap either secret keys or private keys or both is allowed and enabled, and
   - The module and the external entity share an appropriate private key or secret key needed to protect the confidentiality of the key material.

3.9. Firmware Security

The Firmware Security Policy assumes that any firmware images loaded in conformance with the policy have been verified by SafeNet Canada to ensure that the firmware will function correctly. The policy applies to initial firmware loading and subsequent firmware updates.

The module shall not allow external software[4] to be loaded inside its boundary. Only properly formatted firmware may be loaded. The communication of initial or updated firmware to a target module shall be initiated by a SafeNet Canada module dedicated to that function. Firmware shall be digitally signed using the SafeNet Canada Manufacturing signature key and encrypted using a secret key that may be derived by the receiving module for decryption. The unencrypted firmware must not be visible outside the module before, during and after the loading operation.

The firmware shall provide mechanisms to ensure its own integrity and to ensure the integrity of any permanent security-critical data stored within the module.

3.10. Physical Security

The K3 cryptographic module is a multi-chip embedded module as defined by FIPS PUB 140-2 section 4.5. It is enclosed in a strong enclosure that provides tamper-evidence, and detection and response features. Any tampering that might compromise the module's security is detectable by visual inspection of the physical integrity of the module. Attempts to remove the enclosure are detected and the module responds by entering an inoperative state and erasing all plaintext sensitive data from volatile and non-volatile memory.

The module's physical design also resists visual inspection of the device design, physical probing of the device and attempts to access sensitive data on individual components of the device.

3.11. Fault Tolerance

If power is lost to the module for whatever reason, the module shall, at a minimum, maintain itself in a state that it can be placed back into operation when power is restored without compromise of its functionality or permanently stored data.

The module shall maintain its secure state[5] in the event of data input/output failures. When data input/output capability is restored the module will resume operation in the state it was prior to the input/output failure.

3.12. Backup And Recovery

[4] External software means any form of executable code that has been generated by anyone other than SafeNet Canada and has not been properly formatted and signed as a legitimate SafeNet Canada firmware image.

[5] A secure state is one in which either the K3 is operational and its security policy enforcement is functioning correctly, or it is not operational and all sensitive material is stored in a cryptographically protected form on the K3.

The module shall provide the capability to securely backup a partition by the Partition User to a backup PC Card token using the Cloning function. The backup token is then stored in a physically-secured cabinet or safe and retrieved to allow recovery to a second module and partition.

The backup token uses the same firmware as the K3. All sensitive keys stored permanently on the backup token are either encrypted under the SO's Master Key (SMK) or under the User's Storage Key (USK), depending on the key's owner. All sensitive, permanent cryptographic material used for external cryptographic services is stored on the backup token in one of the flash memory data blocks. Non-sensitive, permanent keys (i.e., public keys) stored on the backup token are stored in plain-text in flash.

3.13. Mitigation of Other Attacks

Timing attacks are mitigated directly by the module through the use of hardware accelerator chips for modular exponentiation operations. The use of hardware acceleration ensures that all RSA signature operations complete in very nearly the same time, therefore making the analysis of timing differences irrelevant. RSA blinding may also be selected as an option to mitigate this type of attack.

APPENDIX A. Cryptographic Algorithms Support

FIPS-approved algorithms are shown in bold lettering.

Encrypt/Decrypt:

- **DES-ECB**
- **DES-CBC**
- **3-DES-ECB**
- **3-DES-CBC**
- **AES ECB**
- **AES CBC**
- RC2-ECB
- RC2-CBC
- RC4
- RC5-ECB
- RC5-CBC
- CAST-ECB
- CAST-CBC
- CAST3-ECB
- CAST3-CBC
- CAST5-ECB
- CAST5-CBC
- RSA X-509
- SEED

Digest:

- **SHA-1**
- MD2
- MD5

Sign/Verify:

- **RSA -1024**
- **RSA -2048**
- **DSA**
- **DES-MAC**
- **3-DES-MAC**
- AES MAC
- **HMAC-SHA1**
- RC2-MAC
- RC5-MAC
- CAST-MAC
- CAST3-MAC
- CAST5-MAC
- SSL3-MD5-MAC
- SSL3-SHA1-MAC
- HMAC-MD5
- KCDSA

Generate Key:

- **DES**
- **double length DES**
- **triple length DES**
- **AES 128, 192, 256 bits**
- RC2
- RC4
- RC5
- CAST
- CAST3
- CAST5
- SEED
- PBE-MD2-DES
- PBE-MD5-DES

- PBE-MD5-CAST
- PBE-MD5-CAST3
- PBE-SHA-1-CAST5
- GENERIC-SECRET
- SSL PRE-MASTER

Generate Key Pair:

- **RSA-1024**
- **RSA-2048**
- **RSA-4096**
- **DSA-1024**
- DH-1024
- KCDSA

Wrap Symmetric Key Using Symmetric Algorithm:

- **DES-ECB**
- **3-DES-ECB**
- **AES ECB**
- RC2-ECB
- CAST-ECB
- CAST3-ECB
- CAST5-ECB

Wrap Symmetric Key Using Asymmetric Algorithm:

- **RSA-1024**
- **RSA-2048**
- **RSA 4096**

Wrap Asymmetric Key Using Symmetric Algorithm:

- **3-DES-CBC**
- **AES-CBC**

Unwrap Symmetric Key With Symmetric Algorithm:

- **DES-ECB**
- **3-DES-ECB**
- **AES ECB**
- RC2-ECB
- CAST-ECB
- CAST3-ECB
- CAST5-ECB

Unwrap Symmetric Key With Asymmetric Algorithm:

- **RSA-1024**
- **RSA-2048**
- **RSA-4096**

Unwrap Asymmetric Key With Symmetric Algorithm:

- **DES-CBC**
- **3-DES-CBC**
- **AES-CBC**
- CAST-CBC
- CAST3-CBC
- CAST5-CBC

APPENDIX B. SECURITY POLICY CHECKLIST TABLES

Table B-1 Roles and Required Identification and Authentication

Role	Type of Authentication	Authentication Data
Security Officer	Identity-based	Authentication token (PED Key – one per module) plus optional PED PIN
Partition User	Identity-based	Authentication token (PED Key – one per user) plus optional PED PIN, plus optional Challenge Secret
Public User	Not required	N/A

Table B-2 Strengths of Authentication Mechanisms

Authentication Mechanism	Strength of Mechanism
PED Key	48 byte random authentication data store on PED Key
Challenge Secret	16 character random string

Table B-3 Services Authorized for Roles

Role	Authorized Services
Security Officer	Show Status, Self-test, Initialize Module, Configure Module Policy, Create Partition, Configure Partition Policy, HSM Backup and Restore
Partition User	Show Status, Self-test, Key and Key Pair Generation, Symmetric Encrypt/Decrypt, Asymmetric Signature/Verification, Symmetric & Asymmetric Key Wrap/Unwrap, Store Data Object, Read Data Object, Partition Backup and Restore
Public User	Show Status, Self-test

Table B-4 Access Rights within Services

Service	Cryptographic Keys and CSPs	Role	Type(s) of Access
Show Status	N/A	All	N/A
Self-test	N/A	All	N/A
Initialize Module	Authentication data via trusted path	SO	Write – SO authentication data
Configure Module Policy	Authentication data via trusted path	SO	Use[6]
Create Partition	Authentication data via trusted path	SO	Write – User authentication data
Configure Partition Policy	Authentication data via trusted path	SO	Use
Key and Key Pair Generation	Symmetric keys, asymmetric key pairs	User	Write
Symmetric Encrypt/Decrypt	Symmetric keys	User	Use
Asymmetric Signature	RSA, DSA private keys	User	Use
Asymmetric Verification	RSA, DSA public keys	User	Use
Symmetric Key Wrap/ Unwrap	Symmetric with RSA Symmetric with Symmetric ECB mode	User	Use
Asymmetric Key Wrap/ Unwrap	Asymmetric with Symmetric CBC mode	User	Use
Symmetric Key Mask/ Unmask	Symmetric with AES 256	User	Use
Asymmetric Key Mask/ Unmask	Symmetric with AES 256	User	Use
Store Data Object	Non-cryptographic data	User	Write
Read Data Object	Non-cryptographic data	User	Read
Backup Keys	Symmetric keys, asymmetric key pairs	User	Transfer[7]

[6] Use means access to key material for use in performing a cryptographic operation. The key material is never visible.
[7] Transfer means moving a key using the cloning protocol from one crypto module.