THALES

SafeNet Luna PCIe HSM 7.3

ADMINISTRATION GUIDE

Document Information

Product Version	7.3
Document Part Number	007-013578-005
Release Date	13 December 2019

Revision History

Revision	Date	Reason
Rev. A	13 December 2019	Initial release.

damages or any damages whatsoever including but not limited to damages resulting from loss of use, data, profits, revenues, or customers, arising out of or in connection with the use or performance of information contained in this document.

Thales does not and shall not warrant that this product will be resistant to all possible attacks and shall not incur, and disclaims, any liability in this respect. Even if each product is compliant with current security standards in force on the date of their design, security mechanisms' resistance necessarily evolves according to the state of the art in security and notably under the emergence of new attacks. Under no circumstances, shall Thales be held liable for any third party actions and in particular in case of any successful attack against systems or equipment incorporating Thales products. Thales disclaims any liability with respect to security for direct, indirect, incidental or consequential damages that result from any use of its products. It is further stressed that independent testing and verification by the person using the product is particularly encouraged, especially in any application in which defective, incorrect or insecure functioning could result in damage to persons or property, denial of service, or loss of privacy.

Regulatory Compliance

This product complies with the following regulatory regulations. To ensure compliancy, ensure that you install the products as specified in the installation instructions and use only Thales-supplied or approved accessories.

USA, FCC

This equipment has been tested and found to comply with the limits for a “Class B” digital device, pursuant to part 15 of the FCC rules.

Canada

This class B digital apparatus meets all requirements of the Canadian interference-causing equipment regulations.

Europe

This product is in conformity with the protection requirements of EC Council Directive 2014/30/EU. This product satisfies the CLASS B limits of EN55032.

CONTENTS

Preface: About the Administration Guide 10
Customer Release Notes 11
Audience 11
Document Conventions 11
Support Contacts 13

Chapter 1: Audit Logging 14
Audit Logging Overview 14
Audit limitations and Controlled tamper recovery state 16
The Audit Role 16
Audit Log Records 18
Audit Log Message Format 19
Configuring and Using Audit Logging 22
Configuring Audit Logging 22
Exporting the Audit Logging Secret and Importing to a Verifying HSM 24
Reading the Audit Log Records 25
Audit Role Authentication Considerations 26
Audit Logging General Advice and Recommendations 26
Audit Log Categories and HSM Events 27

Chapter 2: Backup and Restore 34
Backup & Restore Overview 34
Connect 34
Source and Target - full or partial 36
PED or Password 37
Remote Backup and Restore 37
About HSM Backup - Local and Remote 37
The Backup HSM 38
Local Backup of co-located HSMs 38
Local Backup of a Distant SafeNet HSM 39
Preparing (configuring) for Remote Backup with Remote PED 39
Remote Backup 40
About HSM Backup - Local and Remote 41
The Backup HSM 41
Local Backup of co-located HSMs 42
Local Backup of a Distant SafeNet HSM 42
Preparing (configuring) for Remote Backup with Remote PED 42
Remote Backup 43
Backup HSM Installation, Storage, and Maintenance 44
Backup your HSM 50
Disconnecting SafeNet Luna Backup HSM or SafeNet Luna USB HSM 51

Remote Backup Service 51
Prepare RBS to Support Backup / Restore 59
Backup your HSM Partition Remotely 61
Remote Backup Requirements 62
Assumptions 62
RBS Remote Backup with Single Remote PED on Windows 64
Restore to a SafeNet Luna PCIe HSM Slot 68
Restore your HSM Partition Locally 74
Restore your HSM Partition from Token 74
Troubleshooting and Frequently Asked Questions 77
Troubleshooting "token not in factory reset state" Error 78
Backup HSM Battery Questions 79
Should I take the battery out when storing the HSM in a safe? 80
If the battery is out, what happens? 80
If the battery dies during operation, will I lose my key material? Will corruption occur? 80
Where can I get a spare/replacement battery? 80
How do I know if the battery is dead or about to die? Can I check the status of the battery? 80
What must I do to recover function, and access to my key material, after battery removal/discharge? 81

Chapter 3: Capabilities and Policies 82
HSM Capabilities and Policies 82
Partition Capabilities and Policies 87
Policy Templates 93
Creating a Policy Template 94
Editing a Policy Template 94
Applying a Policy Template 97

Chapter 4: Configuration File Summary 99

Chapter 5: Decommissioning, Zeroizing, Re-imaging, or Resetting an HSM to Factory Conditions 117
Zeroization 117
Decommissioning the HSM Card 118
Disabling Decommissioning 119
Resetting to Factory Condition 119
Comparing Zeroize, Decommission, and Factory Reset 119
End of Service and Disposal 120
Comparison of Destruction/Denial Actions 121
RMA and Shipping Back to Thales Group 123

Chapter 6: High-Availability Groups 124
How HA Works 126
Performance 126
Load Balancing 127
Key Replication 128
Failover 129
Recovery 130

Standby Members 131
Process Interaction 132
Application Object Handles 132
Example: Database Encryption 133
Planning Your HA Group Deployment 134
HSM and Partition Prerequisites 134
Sample Configuration 135
Setting Up an HA Group 136
Verifying an HA Group 139
Setting an HA Group Member to Standby 142
Configuring HA Auto-Recovery 144
Enabling/Disabling HA Only Mode 145
HA Logging 146
Configuring HA Logging 146
HA Log Messages 147
Managing Your HA Groups 150
Adding/Removing an HA Group Member 150
Manually Recovering a Failed HA Group Member 154
Replacing an HA Group Member 155
Deleting an HA Group 158
HA Troubleshooting 158
Administration Tasks on HA Groups 158
Unique Object IDs (OUID) 158
Client-Side Failures 159
Effect of PED Operations 159

Chapter 7: HSM Initialization 160
Initializing a New or Factory-reset HSM 161
Re-initializing an Existing, Non-factory-reset HSM 163
PED-authenticated HSM Initialization Example 163
Password-authenticated HSM Initialization Example 169

Chapter 8: HSM Status Values 170

Chapter 9: Keys In Hardware vs. Private Key Export 172
Cloning Mode 172
Key Export Mode 173
No Backup Mode 174

Chapter 10: Partitions 176
About HSM Partitions 176
Configured and Registered Client Using an HSM Partition 177
Activation and Auto-Activation on PED-Authenticated Partitions 178
Auto-Activation 182
Security of Your Partition Challenge 182
Removing Partitions 184
Frequently Asked Questions 185

Chapter 11: PED Authentication 187
PED Authentication Architecture 187
Comparing Password and PED Authentication 188
PED Keys 189
PED Key Types and Roles 189
Shared PED Key Secrets 191
M of N Split Secrets (Quorum) 192
SafeNet Luna PED Hardware Functions 193
Physical Features 193
Keypad Functions 194
Modes of Operation 195
Local PED Setup 196
Local PED Troubleshooting 197
About Remote PED 198
Remote PED Architecture 198
PEDserver-PEDclient Communications 201
Remote PED Setup 202
Initializing the Remote PED Vector (RPV) and Creating the Orange PED Key 202
Installing PEDserver and Setting Up the Remote Luna PED 203
Opening a Remote PED Connection 205
Ending or Switching the Remote PED Connection 207
Remote PED Troubleshooting 208
PED Key Management 211
Creating PED Keys 212
Performing PED Authentication 217
Consequences of Losing PED Keys 218
Identifying a PED Key Secret 221
Duplicating Existing PED Keys 222
Changing a PED Key Secret 222
PEDserver and PEDclient 225
The PEDserver Utility 225
The PEDclient Utility 225
pedclient 226
pedclient mode assignid 228
pedclient mode config 229
pedclient mode deleteid 231
pedclient mode releaseid 232
pedclient mode setid 233
pedclient mode show 234
pedclient mode start 235
pedclient mode stop 237
pedclient mode testid 238
pedserver 239
pedserver appliance 240
pedserver appliance delete 241
pedserver appliance list 242
pedserver appliance register 243

pedserver mode 244
pedserver mode config 245
pedserver mode connect 247
pedserver mode disconnect 248
pedserver mode show 249
pedserver mode start 251
pedserver mode stop 253
pedserver regen 255

Chapter 12: Performance Monitoring 256

Chapter 13: Security in Operation 257
Security Effects of Administrative Actions 257

Chapter 14: Secure Transport Mode 262
Placing an HSM Into Secure Transport Mode 264
Recovering an HSM From Secure Transport Mode 264

Chapter 15: Slot Numbering and Behavior 266
Order of Occurrence for Different SafeNet Luna HSMs 266
Settings Affecting Slot Order 267
Effects of Settings on Slot List 267
Effects of New Firmware on Slot Login State 268

Chapter 16: SNMP Monitoring 269
Overview and Installation 269
MIB 269
SafeNet SNMP Subagent 269
The SafeNet Chrysalis-UTSP MIB 271
The SafeNet Luna HSM MIB 272
hsmPolicyTable 275
hsmPartitionPolicyTable 275
hsmClientRegistrationTable 276
hsmClientPartitionAssignmentTable 276
SNMP output compared to SafeNet tools output 277
Frequently Asked Questions 280

Chapter 17: Tamper Events 281
Recovering from a Tamper Event 282

Chapter 18: Troubleshooting 284
General Troubleshooting Tips 284
System Operational and Error Messages 284
Extra slots that say "token not present"? 284
Error: 'hsm update firmware' failed. (10A0B : LUNA_RET_OPERATION_RESTRICTED) when attempting to perform hsm update firmware 285

KR_ECC_POINT_INVALID Error when decrypting a file encrypted from BSAFE through ECIES using ECC key with any of the curves from the x9_t2 section 285
Slow/interrupted response from the HSM, and the "hsm show" command shows LUNA_RET_SM_SESSION_REALLOC_ERROR 285
Low Battery Message 286
Keycard and Token Return Codes 286
Library Codes 304
Vendor-Defined Return Codes 308

Chapter 19: Updates and Upgrades 315
Update Considerations 315
Valid Update Paths 315
FIPS-Validated Firmware Versions 316
Recommended Minimum Versions 316
Version Dependencies by Feature 317
Updating the SafeNet Luna HSM Client 318
Updating the SafeNet Luna PCIe HSM or SafeNet Luna Backup HSM Firmware 318
Changing the Firmware Upgrade Permissions (Linux only) 319
Rolling Back the SafeNet Luna HSM Firmware 319
Upgrading HSM Capabilities 320

Chapter 20: Users and Roles 321
HSM Roles and Procedures 322
HSM Security Officer (SO) 322
Auditor (AU) 323
Logging In as HSM Security Officer 323
Logging In as Auditor 323
Changing a Role Credential 324
Partition Roles and Procedures 325
Initializing the Crypto Officer and Crypto User Roles 326
Logging In to the Application Partition 327
Changing a Role Credential 328
Resetting the Crypto Officer or Crypto User Credential 329
Failed Login Attempts 329

PREFACE: About the Administration Guide

This document describes the operational and administrative tasks you can perform to maintain the functionality and efficiency of your HSMs. It contains the following chapters:

- "Audit Logging" on page 14
- "Backup and Restore" on page 34
- "Capabilities and Policies" on page 82
- "Configuration File Summary" on page 99
- "Decommissioning, Zeroizing, Re-imaging, or Resetting an HSM to Factory Conditions" on page 117
- "High-Availability Groups" on page 124
- "HSM Initialization" on page 160
- "HSM Status Values" on page 170
- "Keys In Hardware vs. Private Key Export" on page 172
- "Partitions" on page 176
- "PED Authentication" on page 187
- "Performance Monitoring" on page 256
- "Security Effects of Administrative Actions" on page 257
- "Secure Transport Mode" on page 262
- "Slot Numbering and Behavior" on page 266
- "SNMP Monitoring" on page 269
- "Tamper Events" on page 281
- "Troubleshooting" on page 284
- "Updates and Upgrades" on page 315
- "Users and Roles" on page 321

This preface also includes the following information about this document:

- "Customer Release Notes" on the next page
- "Audience" on the next page
- "Document Conventions" on the next page
- "Support Contacts" on page 13

For information regarding the document status and revision history, see "Document Information" on page 2.

Customer Release Notes

The customer release notes (CRN) provide important information about this release that is not included in the customer documentation. Read the CRN to fully understand the capabilities, limitations, and known issues for this release. You can view or download the latest version of the CRN from the Technical Support Customer Portal at https://supportportal.gemalto.com.

Audience

This document is intended for personnel responsible for maintaining your organization's security infrastructure. This includes SafeNet Luna HSM users and security officers, key manager administrators, and network administrators.

All products manufactured and distributed by Thales Group are designed to be installed, operated, and maintained by personnel who have the knowledge, training, and qualifications required to safely perform the tasks assigned to them. The information, processes, and procedures contained in this document are intended for use by trained and qualified personnel only.

It is assumed that the users of this document are proficient with security concepts.

Document Conventions

This document uses standard conventions for describing the user interface and for alerting you to important information.

Notes

Notes are used to alert you to important or helpful information. They use the following format:

> **NOTE** Take note. Contains important or helpful information.

Cautions

Cautions are used to alert you to important information that may help prevent unexpected results or data loss. They use the following format:

> **CAUTION!** Exercise caution. Contains important information that may help prevent unexpected results or data loss.

Warnings

Warnings are used to alert you to the potential for catastrophic data loss or personal injury. They use the following format:

> ****WARNING** Be extremely careful and obey all safety and security measures. In this situation you might do something that could result in catastrophic data loss or personal injury.**

Command syntax and typeface conventions

Format	Convention
bold	The bold attribute is used to indicate the following: > Command-line commands and options (Type **dir /p**.) > Button names (Click **Save As**.) > Check box and radio button names (Select the **Print Duplex** check box.) > Dialog box titles (On the **Protect Document** dialog box, click **Yes**.) > Field names (**User Name**: Enter the name of the user.) > Menu names (On the **File** menu, click **Save**.) (Click **Menu** > **Go To** > **Folders**.) > User input (In the **Date** box, type **April 1**.)
italics	In type, the italic attribute is used for emphasis or to indicate a related document. (See the *Installation Guide* for more information.)
<variable>	In command descriptions, angle brackets represent variables. You must substitute a value for command line arguments that are enclosed in angle brackets.
[**optional**] [<optional>]	Represent optional **keywords** or <variables> in a command line description. Optionally enter the keyword or <variable> that is enclosed in square brackets, if it is necessary or desirable to complete the task.
{**a**|**b**|**c**} {<a>|<b>|<c>}	Represent required alternate **keywords** or <variables> in a command line description. You must choose one command line argument enclosed within the braces. Choices are separated by vertical (OR) bars.
[**a**|**b**|**c**] [<a>|<b>|<c>]	Represent optional alternate keywords or variables in a command line description. Choose one command line argument enclosed within the braces, if desired. Choices are separated by vertical (OR) bars.

Support Contacts

If you encounter a problem while installing, registering, or operating this product, please refer to the documentation before contacting support. If you cannot resolve the issue, contact your supplier or Thales Customer Support.

Thales Customer Support operates 24 hours a day, 7 days a week. Your level of access to this service is governed by the support plan arrangements made between Thales and your organization. Please consult this support plan for further information about your entitlements, including the hours when telephone support is available to you.

Customer Support Portal

The Customer Support Portal, at https://supportportal.thalesgroup.com, is where you can find solutions for most common problems. The Customer Support Portal is a comprehensive, fully searchable database of support resources, including software and firmware downloads, release notes listing known problems and workarounds, a knowledge base, FAQs, product documentation, technical notes, and more. You can also use the portal to create and manage support cases.

> **NOTE** You require an account to access the Customer Support Portal. To create a new account, go to the portal and click on the **REGISTER** link.

Telephone

The support portal also lists telephone numbers for voice contact (Contact Us).

CHAPTER 1: Audit Logging

This chapter describes how to use audit logging to provide security audits of HSM activity. It contains the following sections:

> "Audit Logging Overview" below
> "Configuring and Using Audit Logging" on page 22
> "Audit Logging General Advice and Recommendations" on page 26
> "Audit Log Categories and HSM Events" on page 27

Audit Logging Overview

Each event that occurs on the HSM can be recorded in the HSM event log, allowing you to audit your HSM usage. The HSM event log is viewable and configurable only by the **audit** user role. This audit role is disabled by default and must be explicitly enabled.

Types of events included in the logs

The events that are included in the log is configurable by the audit role. The types of events that can be logged include the following:

> log access attempts (logins)
> log HSM management (init/reset/etc)
> key management events (key create/delete)
> asymmetric key usage (sig/ver)
> first asymmetric key usage only (sig/ver)
> symmetric key usage (enc/dec)
> first symmetric key usage only (enc/dec)
> log messages from CA_LogExternal
> log events relating to log configuration

Each of these events can be logged if they fail, succeed, or both.

Event log storage

When the HSM logs an event, the log is stored on the HSM. The audit user cannot view these log entries. Before a log can be viewed, it must be rotated. Log rotation saves the log entries on the HSM to the local file system, where they can be viewed. Log records are HMACed using an audit log secret to ensure their authenticity. The audit log secret is unique to the HSM where the log was created, and is required to view the HSM event logs. The secret can be exported, allowing you to view and verify the logs on another HSM.

Event logging impacts HSM performance
Each audit log record generated requires HSM resources. Configuring event logging to record most, or all, events may have an impact on HSM performance. You may need to adjust your logging configuration to provide adequate logging without significantly affecting performance. By default, only critical events are logged, imposing virtually no load on the HSM.

Audit Logging Features

The following list summarizes the functionality of the audit logging feature:

- Log entries originate from the SafeNet Luna PCIe HSM - the feature is implemented via HSM firmware (rather than in the library) for maximum security.
- Log origin is assured.
- Logs and individual records can be validated by any SafeNet Luna PCIe HSM that is a member of the same domain.
- Audit Logging can be performed on password-authenticated (FIPS 140-2 level 2) and PED-authenticated (FIPS 140-2 level 3) configurations, but these configurations may not validate each other's logs - see the "same domain" requirement, above.
- Each entry includes the following:
  - When the event occurred
  - Who initiated the event (the authenticated entity)
  - What the event was
  - The result of the logging event (success, error, etc.)
- Multiple categories of audit logging are supported, configured by the audit role.
- Audit management is a separate role - the role creation does not require the presence or co-operation of the SafeNet Luna PCIe HSM SO.
- The category of audit logging is configurable by (and only by) the audit role.
- Audit log integrity is ensured against the following:
  - Truncation - erasing part of a log record
  - Modification - modifying a log record
  - Deletion - erasing of the entire log record
  - Addition - writing of a fake log record
- Log origin is assured.
- The following critical events are logged unconditionally, regardless of the state of the audit role (initialized or not):
  - Tamper
  - Decommission
  - Zeroization
  - SO creation
  - Audit role creation

Audit limitations and Controlled tamper recovery state

The following conditions apply when HSM Policy "48: Do controlled tamper recovery" is enabled (default setting).

- Auditor (the Audit role) cannot verify the integrity of audit logs until after recovery from tamper.
- Auditor cannot be initialized when the HSM is in controlled tamper recovery state.
- Existing Audit role can login when in controlled tamper recovery state.
- Existing Audit role cannot make audit config changes when in controlled tamper recovery state.
- Existing Audit role cannot export the audit secret when in controlled tamper recovery state.

The Audit Role

A SafeNet Luna PCIe HSM Audit role allows complete separation of Audit responsibilities from the Security Officer (SO or HSM Admin), the Partition User (or Owner), and other HSM roles. If the Audit role is initialized, the HSM and Partition administrators are prevented from working with the log files, and auditors are unable to perform administrative tasks on the HSM. As a general rule, the Audit role should be created before the HSM Security Officer role, to ensure that all important HSM operations (including those that occur during initialization), are captured.

Use the LunaCM command **role init -name Auditor** to initialize the audit role, as described in "role init" on page 1.

Password-authenticated HSMs

For SafeNet Luna PCIe HSMs with Password Authentication, the auditor role logs into the HSM to perform their activities using a password. After initializing the Audit role on a password-authenticated HSM, log in as the Auditor and set the domain (see "role setdomain" on page 1 for the command syntax). This step is required before setting logging parameters or the log filepath, or importing/exporting audit logs.

PED-authenticated HSMs

For SafeNet Luna PCIe HSMs with PED Authentication, the auditor role logs into the HSM to perform their activities using the Audit (white) PED key.

Role Initialization

Creating the Audit role (and imprinting the white PED key for PED-authenticated HSMs) does not require the presence or cooperation of the HSM SO.

Audit Role Available Commands

In LunaCM, all commands are visible to the person who launches the utility, and some can be used without specific authentication to the HSM, such as view/show/list commands, which might be classified as "monitoring" functions. Attempts to run operational or administrative commands that need role-specific authentication, without that authentication, result in an error message. The Audit role has a limited set of operations available to it, on the HSM, which constitutes any of the generally accessible monitoring commands, plus everything under the "audit" heading.

```
lunacm:>audit


  The following sub commands are available:
```

```
  Command         Short    Description
  ----------------------------------

  verify          v        Verify a block of log messages
  config          c        Configure audit parameters
  export          e        Read the wrapped log secret from the HSM
  import          m        Import the wrapped log secret to the HSM
  time            t        Sync HSM time to host, or get HSM time
  status          s        Show status of logging subsystem
  logmsg          logm     Write a message to the HSM's log

  Syntax: audit <sub command>


Command Result : No Error
```

Anyone accessing the computer and running LunaCM can see the above commands, but cannot run them if they do not have the "audit" role authentication (password or PED key, as appropriate).

What is important is not the role you can access on the computer (a named user, admin, root), but the role you can access within the HSM.

Audit Log Secret

The HSM creates a log secret unique to the HSM, computed during the first initialization after manufacture. The log secret resides in flash memory (permanent, non-volatile memory), and is used to create log records that are sent to a log file. Later, the log secret is used to prove that a log record originated from a legitimate HSM and has not been tampered with.

Log Secret and Log Verification

The 256-bit log secret which is used to compute the HMACs is stored in the parameter area on the HSM. It is set the first time an event is logged. It can be exported from one HSM to another so that a particular sequence of log messages can be verified by the other HSM. Conversely, it can be imported from other HSMs for verification purpose.

To accomplish cross-HSM verification, the HSM generates a key-cloning vector (KCV, a.k.a. the Domain key) for the audit role when it is initialized. The KCV can then be used to encrypt the log secret for export to the HOST.

To verify a log that was generated on another HSM, assuming it is in the same domain, we simply import the wrapped secret, which the HSM subsequently decrypts; any records that are submitted to the host for verification will use this secret thereafter.

When the HSM exports the secret, it calculates a 32-bit checksum which is appended to the secret before it is encrypted with the KCV.

When the HSM imports the wrapped secret, it is decrypted, and the 32-bit checksum is calculated over the decrypted secret. If this doesn't match the decrypted checksum, then the secret that the HSM is trying to import comes from a system on a different domain, and an error is returned.

To verify a log generated on another HSM, in the same domain, the host passes to the target HSM the wrapped secret, which the target HSM subsequently decrypts; any records submitted to the target HSM for verification use this secret thereafter.

Importing a log secret from another HSM does not overwrite the target log secret because the operation writes the foreign log secret only to a separate parameter area for the wrapped log secret.

> **CAUTION!** Once an HSM has imported a wrapped log secret from another HSM, it must export and then re-import its own log secret in order to verify its own logs again.

Audit Log Records

A log record consists of two fields – the log message and the HMAC for the previous record. When the HSM creates a log record, it uses the log secret to compute the SHA256-HMAC of all data contained in that log message, plus the HMAC of the previous log entry. The HMAC is stored in HSM flash memory. The log message is then transmitted, along with the HMAC of the previous record, to the host. The host has a logging daemon to receive and store the log data on the host hard drive.

For the first log message ever returned from the HSM to the host there is no previous record and, therefore, no HMAC in flash. In this case, the previous HMAC is set to zero and the first HMAC is computed over the first log message concatenated with 32 zero-bytes. The first record in the log file then consists of the first log message plus 32 zero-bytes. The second record consists of the second message plus HMAC1 = HMAC (message1 || 0x0000). This results in the organization shown below.

MSG 1	HMAC 0
	. . .
MSG n-1	HMAC n-2
MSG n	HMAC n-1
. . .	
MSG n+m	HMAC n+m-1
MSG n+m+1	HMAC n+m
. . .	
MSG end	HMAC n+m-1

Recent HMAC in NVRAM	HMAC end

To verify a sequence of *m* log records which is a subset of the complete log, starting at index *n*, the host must submit the data illustrated above. The HSM calculates the HMAC for each record the same way as it did when the record was originally generated, and compares this HMAC to the value it received. If all of the calculated HMACs match the received HMACs, then the entire sequence verifies. If an HMAC doesn't match, then the associated record and all following records can be considered suspect. Because the HMAC of each message depends on the HMAC of the previous one, inserting or altering messages would cause the calculated HMAC to be invalid.

The HSM always stores the HMAC of the most-recently generated log message in flash memory. When checking truncation, the host would send the newest record in its log to the HSM; and, the HSM would compute the HMAC and compare it to the one in flash. If it does not match, then truncation has occurred.

Audit Log Message Format

Each message is a fixed-length, comma delimited, and newline-terminated string. The table below shows the width and meaning of the fields in a message.

Offset	Length (Chars)	Description
0	10	Sequence number
10	1	Comma
11	17	Timestamp
28	1	Comma
29	256	Message text, interpreted from raw data
285	1	Comma
286	64	HMAC of previous record as ASCII-HEX
350	1	Comma
351	96	Data for this record as ASCII-HEX (raw data)
447	1	Newline '\n'

The raw data for the message is stored in ASCII-HEX form, along with a human-readable version. Although this format makes the messages larger, it simplifies the verification process, as the HSM expects to receive raw data records.

Example

The following example shows a sample log record. It is separated into multiple lines for readability even though it is a single record. Some white spaces are also omitted.

```
38,12/08/13 15:30:50,session 1 Access 2147483651:22621 operation LUNA_CREATE_CONTAINER
returned LUNA_RET_SM_UNKNOWN_TOSM_STATE(0x00300014) (using PIN (entry=LUNA_ENTRY_DATA_AREA)),
29C51014B6F131EC67CF48734101BBE301335C25F43EDF8828745C40755ABE25,
2600001003600B00EA552950140030005D5800000300008001000000000000000000000000000000000000000000
```

The sequence number is "`38`". The time is "`12/08/13 15:30:50`".

The log message is "`session 1 Access 2147483651:22621 operation LUNA_CREATE_CONTAINER returned LUNA_RET_SM_UNKNOWN_TOSM_STATE(0x00300014) (using PIN (entry=LUNA_ENTRY_DATA_AREA))`".

In the message text, the "who" is the session identified by "`session 1 Access 2147483651:22621`" (the application is identified by the access ID major = 2147483651, minor = 22621).

The “what” is “LUNA_CREATE_CONTAINER”.

The operation status is “LUNA_RET_SM_UNKNOWN_TOSM_STATE(0x00300014)”.

The HMAC of previous record is
“29C51014B6F131EC67CF48734101BBE301335C25F43EDF8828745C40755ABE25”.

The remainder is the raw data for this record as ASCII-HEX.

> The “who” is LunaSH session “session 1 Access 2147483651:22621”
(identified by the lunash access ID major = 2147483651, minor = 22621).

> The “what” is “LUNA_CREATE_CONTAINER”.

> The operation status is “LUNA_RET_SM_UNKNOWN_TOSM_STATE(0x00300014)”.

NOTE Log Rotation Categories, Rotation Intervals, and other Configurable Factors are covered here in the *Administration Guide*. Command syntax is in the *Command Reference Guide*.

Timestamping

The HSM has an internal real-time clock (RTC). The RTC does not have a relevant time value until it is synchronized with the HOST system time. Because the HSM and the host time could drift apart over time, periodic re-synchronization is necessary. Only an authenticated Auditor is allowed to synchronize the time.

Time Reported in Log

When you perform **audit time get**, you might see a variance of a few seconds between the reported HSM time and the Host time. Any difference up to five seconds should be considered normal, as the HSM reads new values from its internal clock on a five-second interval. So, typically, Host time would show as slightly ahead.

Log Capacity

The log capacity of SafeNet Luna PCIe HSMs varies depending upon the physical memory available on the device.

The HSM has approximately 16 MB available for Audit logging (or more than 200,000 records, depending on the size/content of each record).

The normal function of Audit logging is to export log entries constantly to the file system. Short-term, within-the-HSM log storage capacity becomes important only in the rare situations where the HSM remains functioning but the file system is unreachable from the HSM.

Log full condition

In the case of a log full condition on the host, most commands will return CKR_LOG_FULL. There are a few exceptions to this, as follows:

> factory reset

> zeroize

> login as audit user

> logout

> open session

- close session
- get audit config
- set audit config

Since the “log full” condition can make the HSM unusable, these commands are required to be able to login as the audit user and disable logging, even if logging for those commands is enabled; and the log is full. All other commands will not execute if their results are supposed to be logged, but can’t be, due to a log full condition.

Configuration Persists Unless Factory Reset is Performed

Audit logging configuration is not removed or reset upon HSM re-initialization or a tamper event. Factory reset or HSM decommission will remove the Audit user and configuration. Logs must be cleared by specific command. Therefore, if your security regime requires decommission at end-of-life, or prior to shipping an HSM, then explicit clearing of HSM logs should be part of that procedure.

This is by design, as part of separation of roles in the HSM. When the Audit role exists, the SO cannot modify the logging configuration, and therefore cannot hide any activity from auditors.

Audit Logging Stops Working if the Current Log File is Deleted

As a general rule, you should not delete a file while it is open and in use by an application. In Linux, deletion of a file is deletion of an inode, but the actual file itself, while now invisible, remains on the file system until the space is cleaned up or overwritten. If a file is in use by an application - such as audit logging, in this case - the application can continue using and updating that file, unaware that it is now in deleted status.

If you delete the current audit log file, the audit logging feature does not detect that and does not create a new file, so you might lose log entries.

The workaround is to restart the **pedclient** daemon, which creates a new log file.

Example

1. You’ve configured audit logging, and the entire audit path is deleted. In Linux, the file isn’t actually deleted until the last reference to the file has been destroyed. Since the pedclient has the file open, logging will continue, because technically the log file still exists. Applications, including the pedclient, will have no idea that anything is wrong.
2. On stopping the pedclient, the log file is deleted. When the pedclient gets started again, the HSM tries to tell the pedclient to use the old path. This path doesn’t exist anymore, so it will not be able to offload log messages. At this point, it starts storing log messages internally. With 16 MB of Flash dedicated to this purpose, that works out to 198,120 messages max. This can actually fill up very quickly, in as little as a few minutes under heavy load.
3. At this point the user must set the audit log path to a valid value. and the HSM will offload all stored log messages to the host. This will take a couple of minutes, during which time the HSM will be unresponsive.
4. Once all messages have been offloaded, normal operation resumes with messages being sent to the host (i.e. not being stored locally).

Configuring and Using Audit Logging

This section describes the procedures required to enable audit logging, configure it to specify what is logged and how often the logs are rotated, and how to copy, verify and read the audit logs. It contains the following information:

> "Configuring Audit Logging" below
> "Exporting the Audit Logging Secret and Importing to a Verifying HSM" on page 24
> "Reading the Audit Log Records" on page 25
> "Audit Role Authentication Considerations" on page 26

Configuring Audit Logging

Configure audit logging using the LunaCM **audit** commands. See "audit" on page 1 in the *LunaCM Command Reference Guide*.

To configure audit logging:

1. Configure the SafeNet Luna PCIe HSM host computer to use network time protocol (NTP).
2. Ensure that the PEDclient service is running:

   See "pedclient mode show" on page 234 and "pedclient mode start" on page 235.
3. Set the slot focus to the HSM administrative partition of the desired HSM:

   lunacm:>**slot set slot** <slotnum>
4. Initialize the Auditor role (you can also use the shortcut **au**):

   lunacm:>**role init -name Auditor**
   - On password-authenticated HSMs, you are prompted for a password.
   - On PED-authenticated HSMs, you are referred to Luna PED, which prompts for a white PED key.
5. Now that the Auditor role exists on the HSM, the auditing function must be configured. However, before you can configure you must log in as the Auditor user (you can also use the shortcut **au**):

   lunacm:> **role login -name au**
   - On password-authenticated HSMs, you are prompted to enter the password for the Auditor user.
   - On PED-authenticated HSMs, you are referred to Luna PED, which prompts for the white PED key for the Auditor user.
6. Set the domain for the Audit role:

   lunacm:> **role setdomain**
7. Synchronize the HSM's clock with the host time (which should also be synchronized with the NTP server) so that all subsequent log records will have a valid and accurate timestamp.

   lunacm:> **audit time sync**
8. Set the filepath where log files are to be saved. You must complete this step before you can start event logging.

   lunacm:> **audit config path**

If you previously configured logging on the HSM and then made changes to your configuration that made that path invalid (such as deleting the path outside of LunaCM or reinstalling the HSM in a different host system), set a valid log path by running **audit config path** before restarting event logging. If the log path is set incorrectly, logs will be stored in the HSM's limited memory and not exported to the file system. Event logs may be lost if the HSM's memory runs out.

9. Configure audit logging to specify what you want to log. You can specify the level of audit appropriate for needs of the organization's policy and the nature of the application(s) using the HSM:

   lunacm:> **audit config evmask** <event_value>

   > **NOTE** Before you configure audit logging, we suggest using **audit config ?** to see all the available options in the configuration process. See also "audit config" on page 1 in the *LunaCM Command Reference Guide*.

   Security audits can generate a very large amount of data, which consumes HSM processing resources, host storage resources, and makes the job of the Auditor quite difficult when it comes time to review the logs. For this reason, ensure that you configure audit logging such that you capture only relevant data, and no more.

   For example, the **First Symmetric Key Usage Only** or **First Asymmetric Key Usage Only** category is intended to assist Auditors to capture the relevant data in a space-efficient manner for high processing volume applications. On the other hand, a top-level Certificate Authority would likely be required, by policy, to capture all operations performed on the HSM but, since it is typically not an application that would see high volumes, configuring the HSM to audit all events would not impose a significant space and/or performance premium in that situation.

   As a further example, the command **audit config evmask all** will log everything the HSM does. This might be useful in some circumstances, but will quickly fill up log files.

10. Configure audit logging to specify how often you want to rotate the logs. Log entries are made within the HSM, and are written to the currently active log file. When a log file reaches the rotation trigger, it is closed, and a new file gets the next log entry. The number of log files grows according to the logging settings and the rotation schedule that you configured. At any time, you can copy files to a remote computer and then clear the originals from the HSM, if you wish to free the space.

    a. Specify the rotation interval. You can rotate the logs hourly, daily, weekly, monthly, or never.

       lunacm:> **audit config interval** <value>

    b. Specify the maximum log file size. When the log reaches the maximum size, it is automatically rotated, regardless of rotation interval:

       lunacm:> **audit config size** <size>

    For example, the commands **audit config interval daily** and **audit config size 4m** would rotate the logs every day, unless they reached a size of 4 Mb first, in which case they would be rotated automatically. The daily rotation would still occur.

    See "audit config" on page 1 in the *LunaCM Command Reference Guide* for additional examples.

Exporting the Audit Logging Secret and Importing to a Verifying HSM

You can export the audit log secret from one HSM and import it to another to allow the first HSM's logs to be viewed and verified on the second. The HSMs must share the same authentication method and Audit cloning domain (password string or red PED key). You can verify logs from a SafeNet Luna PCIe HSM using a SafeNet Luna Network HSM, and vice-versa.

To export the Audit Logging secret from the HSM and import to the verifying HSM:

1. Export the audit logging secret to the user local directory. The file is written to the subdirectory specified by a previous **audit config path** command.

   lunacm:> **audit export file** <filename>

2. Exit LunaCM and list the contents of the **lunalogs** directory to see the filename of the wrapped log secret:

Linux	**ls** <client_install_dir>**/lunalog** `123456 7001347 123456.lws`

Windows	**dir** <client_install_dir>**\lunalog**
	`04/12/2017 03:56 PM <DIR> 123456`
	`04/05/2017 02:35 PM <DIR> 7001347`
	`04/05/2017 02:35 PM 48 123456.lws`

3. Transfer the logging secret to the HSM that will verify the logs. If you are verifying the logs with another locally-installed SafeNet Luna PCIe HSM, skip this step.

   - If you are planning to verify logs with a SafeNet Luna PCIe HSM, use **scp** or **pscp** to transfer the logging secret to the appliance. Provide the audit user's credentials when prompted.

Linux	<client_install_dir>:>**scp** <log_secret_file> **audit@**<hostname_or_IP>**: .**
Windows	<client_install_dir>:>**pscp** <log_secret_file> **audit@**<hostname_or_IP>**: .**

   - If you are planning to verify logs with a SafeNet Luna PCIe HSM installed in a different host computer, you can use **scp**, **pscp**, or other secure means to transfer the logging secret.

Linux	<client_install_dir>:>**scp** <log_secret_file> <user>**@**<hostname_or_IP>**: .**
Windows	<client_install_dir>:>**pscp** <log_secret_file> <user>**@**<hostname_or_IP>**: .**

4. Login to the verifying HSM as the audit user. For this example, we will assume that you have already initialized the HSM audit user role, using the same domain/secret as is associated with the source HSM.

   - If you are using a SafeNet Luna Network HSM, connect via SSH and login to LunaSH as the audit user:

     lunash:>**audit login**

   - If you are using a SafeNet Luna PCIe HSM, open LunaCM and login using the Auditor role:

     lunacm:>**role login -name au**

5. Import the audit logging secret to the HSM.

   - SafeNet Luna Network HSM (LunaSH):

lunash:>**audit secret import -serialtarget** <target_HSM_SN> **-serialsource** <source_HSM_SN> **-file** <log_secret_file>

- SafeNet Luna PCIe HSM (LunaCM):

  lunacm:> **audit import file** <log_secret_file>

6. You can now verify audit log files from the source HSM.
   - SafeNet Luna Network HSM (LunaSH):

     lunash:>**audit log verify -file** <audit_log_filename>**.log**
   - SafeNet Luna PCIe HSM (LunaCM):

     lunacm:> **audit verify file** <audit_log_filename>**.log**

     You might need to provide the full path to the file, depending upon your current environment settings.

> **NOTE** Linux users, if you notice that audit log messages are going to more than one location on your file system, you can edit the /etc/rsyslog.conf file to prevent reporting local3.info messages in /var/log/messages as follows:
>
> ```
> //Log anything (except local3 and mail) of level info or higher.
> *.info;local3.none;mail.none;authpriv.none;cron.none
> /var/log/messages
> ```
>
> The portion highlighted in red stops the duplication of output. This change is optional.

Reading the Audit Log Records

In general, the audit logs are self-explanatory. Due to limitations in the firmware, however, some audit log records required further explanation, as detailed in the following sections:

Determining the serial number of a created partition from the audit log

An audit log entry similar to the following is generated when a partition is created on the HSM:

```
5,12/12/17 16:14:14,S/N 150718 session 1 Access 2147483651:2669 SO container operation LUNA_
CREATE_CONTAINER
returned RC_OK(0x00000000) container=20 (using PIN (entry=LUNA_ENTRY_DATA_AREA))
```

It is not obvious from this entry what the serial number is for the created partition. This information, however, can be derived from the log entry, since the partition serial number is simply a concatenation of the HSM serial number and the partition container number, which are specified in the log entry, as highlighted below:

```
5,12/12/17 16:14:14,S/N 150718 session 1 Access 2147483651:2669 SO container operation LUNA_
CREATE_CONTAINER
returned RC_OK(0x00000000) container=20 (using PIN (entry=LUNA_ENTRY_DATA_AREA))
```

In the example above, the HSM serial number is 150718 and the partition container number is 20. Note that the partition container number is a three-digit number with leading zeros suppressed, so that the actual partition container number is 020. To determine the partition serial number concatenate the two numbers as follows:

```
150718020
```

Use this number to identify the partition in subsequent audit log entries.

Audit Role Authentication Considerations

- The audit role PED key or password is a critical property to manage the audit logs. If that authentication secret is lost, the HSM must be factory reset (that is, zeroize the HSM) in order to initialize the audit role again.
- Multiple bad logins produce different results for the SO and for the audit role, as follows:
  - After 3 bad SO logins, the LUNA_RET_SO_LOGIN_FAILURE_THRESHOLD error is returned and the HSM is zeroized.
  - After 3 bad audit logins, the LUNA_RET_AUDIT_LOGIN_FAILURE_THRESHOLD error is returned, but the HSM is unaffected. If a subsequent login attempt is executed within 30 seconds, the LUNA_RET_AUDIT_LOGIN_TIMEOUT_IN_PROGRESS error is returned. If you wait for more than 30 seconds and try login again with the correct password, the login is successful.

Audit Logging General Advice and Recommendations

The Security Audit Logging feature can produce a significant volume of data. It is expected, however, that Audit Officers will configure it properly for their specific operating environments. The data produced when the feature has been properly configured might be used for a number of reasons, such as:

- Reconstructing a particular action or set of actions (forensics)
- Tracing the actions of an application or individual user (accounting)
- Holding a specific individual accountable for their actions (non-repudiation)

That last point represents the ultimate conclusion of any audit trail – to establish an irrefutable record of the chain of events leading up to a particular incident for the purpose of identifying and holding accountable the individual responsible. Not every organization will want to use security audit to meet the strict requirements of establishing such a chain of events. However, all security audit users will want to have an accurate representation of a particular sequence of events. To ensure that the audit log does contain an accurate representation of events and that it can be readily interpreted when it is reviewed, these basic guidelines should be followed after the audit logging feature has been properly configured:

- Use a shell script to execute the **audit sync** command at least once every 24 hours, provided the host has maintained its connection(s) to its configured NTP server(s).
- Do not allow synchronization with the host's clock if the host has lost connectivity to NTP. This ensures that the HSM's internal clock is not set to a less accurate time than it has maintained internally. In general, the HSM's RTC will drift much less than the host's RTC and will, therefore, be significantly more accurate than the host in the absence of NTP.
- Review logs at least daily and adjust configuration settings if necessary. It is important that any anomalies be identified as soon as possible and that the logging configuration that has been set is effective.
- The audit log records are comma-delimited. We recommend that full use be made of the CSV formatting to import records into a database system or spreadsheet tool for analysis, if an SIEM system is not available.
- The ASCII hex data representing the command and returned values and error code should be examined if an anomaly is detected in log review/analysis. It may be possible to match this data to the HSM's dual-port data. The dual-port, if it is available, will contain additional data that could be helpful in establishing the context surrounding the anomalous event. For example, if an unexpected error occurs it could be possible to identify the trace through the firmware subsystems associated with the error condition. This information

would be needed to help in determining if the error was unexpected but legitimate or if it was forced in an attempt to exploit a potential weakness.

An important element of the security audit logging feature is the 'Log External' function. See the *SDK Reference Guide* for more information. For applications that cannot add this function call, it is possible to use the LunaCM command-line function **audit log external** within a startup script to insert a text record at the time the application is started.

Disk Full

In the event that all the audit disk space is used up, audit logs are written to the HSM's small persistent memory. When the HSM's persistent memory is full, normal crypto commands will fail with "disk full" error.

To resolve that situation, the audit user must:

1. Archive the audit logs on the host side.
2. Move the audit logs to some other location for safe storage.
3. Clear the audit log directory.
4. Restart the logger daemon (**PEDclient**).

To prevent the "disk full" situation, we recommend that the audit user routinely archive the audit logs and clear the audit log directory.

Audit Log Categories and HSM Events

This section provides a summary of the audit log categories and their associated HSM events.

HSM Access

HSM Event	Description
LUNA_LOGIN	C_Login. This event must be allowed to proceed even if the result should be logged but cannot (for example, due to a log full condition).
LUNA_LOGOUT	C_Logout. This event must be allowed to proceed even if the result should be logged but cannot (for example, due to a log full condition).
LUNA_MODIFY_OBJECT	C_SetAttributeValue
LUNA_OPEN_SESSION	C_OpenSession. This event must be allowed to proceed even if the result should be logged but cannot (for example, due to a log full condition).
LUNA_CLOSE_ALL_ SESSIONS	C_CloseAllSessions

HSM Event	Description
LUNA_CLOSE_SESSION	C_CloseSession This event must be allowed to proceed even if the result should be logged but cannot (for example, due to a log full condition).
LUNA_OPEN_ACCESS	CA_OpenApplicationID
LUNA_CLEAN_ACCESS	CA_Restart, CA_RestartForContainer
LUNA_CLOSE_ACCESS	CA_CloseApplicationID
LUNA_LOAD_CUSTOM_ MODULE	CA_LoadModule
LUNA_LOAD_ENCRYPTED_ CUSTOM_MODULE	CA_LoadEncryptedModule
LUNA_UNLOAD_CUSTOM_ MODULE	CA_UnloadModule
LUNA_EXECUTE_CUSTOM_ COMMAND	CA_PerformModuleCall
LUNA_HA_LOGIN	CA_HAGetLoginChallenge, CA_HAAnswerLoginChallenge, CA_HALogin, CA_HAAnswerMofNChallenge, HAActivateMofN

Log External

HSM Event	Description
LUNA_LOG_EXTERNAL	CA_LogExternal

HSM Management

HSM Event	Description
LUNA_ZEROIZE	CA_FactoryReset This event is logged unconditionally.
LUNA_INIT_TOKEN	C_InitToken This event is logged unconditionally.

HSM Event	Description
LUNA_SET_PIN	C_SetPIN
LUNA_INIT_PIN	C_InitPIN
LUNA_CREATE_CONTAINER	CA_CreateContainer
LUNA_DELETE_CONTAINER	CA_DeleteContainer, CA_DeleteContainerWithHandle
LUNA_SEED_RANDOM	C_SeedRandom
LUNA_EXTRACT_CONTEXTS	C_GetOperationState
LUNA_INSERT_CONTEXTS	C_SetOperationState
LUNA_SELF_TEST	C_PerformSelfTest
LUNA_LOAD_CERT	CA_SetTokenCertificateSignature
LUNA_HA_INIT	CA_HAInit
LUNA_SET_HSM_POLICY	CA_SetHSMPolicy
LUNA_SET_DESTRUCTIVE_HSM_POLICY	CA_SetDestructiveHSMPolicy
LUNA_SET_CONTAINER_POLICY	CA_SetContainerPolicy
LUNA_SET_CAPABILITY	Internal, for capability update
LUNA_CREATE_LOGIN_CHALLENGE	CA_CreateLoginChallenge
LUNA_REQUEST_CHALLENGE	CA_SIMInsert, CA_SIMMultiSign
LUNA_PED_INIT_RPV	CA_InitializeRemotePEDVector
LUNA_PED_DELETE_RPV	CA_DeleteRemotePEDVector
LUNA_MTK_LOCK	Internal, for manufacturing
LUNA_MTK_UNLOCK_CHALLENGE	Internal, for manufacturing
LUNA_MTK_UNLOCK_RESPONSE	Internal, for manufacturing
LUNA_MTK_RESTORE	CA_MTKRestore
LUNA_MTK_RESPLIT	CA_MTKResplit

HSM Event	Description
LUNA_MTK_ZEROIZE	CA_MTKZeroize
LUNA_FW_UPGRADE_INIT	CA_FirmwareUpdate
LUNA_FW_UPGRADE_UPDATE	CA_FirmwareUpdate
LUNA_FW_UPGRADE_FINAL	CA_FirmwareUpdate
LUNA_FW_ROLLBACK	CA_FirmwareRollback
LUNA_MTK_SET_STORAGE	CA_MTKSetStorage
LUNA_SET_CONTAINER_SIZE	CA_SetContainerSize

Key Management

HSM Event	Description
LUNA_CREATE_OBJECT	C_CreateObject
LUNA_COPY_OBJECT	C_CopyObject
LUNA_DESTROY_OBJECT	C_DestroyObject
LUNA_DESTROY_MULTIPLE_OBJECTS	CA_DestroyMultipleObjects
LUNA_GENERATE_KEY	C_GenerateKey
LUNA_GENERATE_KEY_PAIR	C_GenerateKeyPair
LUNA_WRAP_KEY	C_WrapKey
LUNA_UNWRAP_KEY	C_UnwrapKey
LUNA_DERIVE_KEY	C_DeriveKey
LUNA_GET_RANDOM	C_GenerateRandom
LUNA_CLONE_AS_SOURCE, LUNA_REPLICATE_AS_SOURCE	CA_CloneAsSource
LUNA_CLONE_AS_TARGET_INIT, LUNA_REPLICATE_AS_TARGET_INIT	CA_CloneAsTargetInit

HSM Event	Description
LUNA_CLONE_AS_TARGET, LUNA_REPLICATE_AS_TARGET	CA_CloneAsTarget
LUNA_GEN_TKN_KEYS	CA_GenerateTokenKeys
LUNA_GEN_KCV	CA_ManualKCV, C_InitPIN, C_InitToken, CA_InitAudit
LUNA_SET_LKCV	CA_SetLKCV
LUNA_M_OF_N_GENERATE	CA_GenerateMofN_Common, CA_GenerateMofN
LUNA_M_OF_N_ACTIVATE	CA_ActivateMofN
LUNA_M_OF_N_MODIFY	CA_ActivateMofN
LUNA_EXTRACT	CA_Extract
LUNA_INSERT	CA_Insert
LUNA_LKM_COMMAND	CA_LKMInitiatorChallenge, CA_LKMReceiverResponse, CA_LKMInitiatorComplete, CA_LKMReceiverComplete.
LUNA_MODIFY_USAGE_COUNT	CA_ModifyUsageCount

Key Usage and Key First Usage

HSM Event	Description
LUNA_ENCRYPT_INIT	C_EncryptInit
LUNA_ENCRYPT	C_Encrypt
LUNA_ENCRYPT_END	C_EncryptFinal
LUNA_DECRYPT_INIT	C_DecryptInit
LUNA_DECRYPT	C_Decrypt
LUNA_DECRYPT_END	C_DecryptFinal
LUNA_DIGEST_INIT	C_DigestInit

HSM Event	Description
LUNA_DIGEST	C_Digest
LUNA_DIGEST_KEY	C_DigestKey
LUNA_DIGEST_END	C_DigestFinal
LUNA_SIGN_INIT	C_SignInit
LUNA_SIGN	C_Sign
LUNA_SIGN_END	C_SignFinal
LUNA_VERIFY_INIT	C_VerifyInit
LUNA_VERIFY	C_Verify
LUNA_VERIFY_END	C_VerifyFinal
LUNA_SIGN_SINGLEPART	C_Sign
LUNA_VERIFY_SINGLEPART	C_Verify
LUNA_WRAP_CSP	CA_CloneMofN_Common
LUNA_M_OF_N_DUPLICATE	CA_DuplicateMofN
LUNA_ENCRYPT_SINGLEPART	C_Encrypt
LUNA_DECRYPT_SINGLEPART	C_Decrypt

Audit Log Management

HSM Event	Description
LUNA_LOG_SET_TIME	CA_TimeSync
LUNA_LOG_GET_TIME	CA_GetTime
LUNA_LOG_SET_CONFIG	CA_LogSetConfig This event must be allowed to proceed even if the result should be logged but cannot (for example, due to a log full condition).

HSM Event	Description
LUNA_LOG_GET_CONFIG	CA_LogGetConfig This event must be allowed to proceed even if the result should be logged but cannot (for example, due to a log full condition).
LUNA_LOG_VERIFY	CA_LogVerify
LUNA_CREATE_AUDIT_CONTAINER **	CA_ InitAudit The event is logged unconditionally.
LUNA_LOG_IMPORT_SECRET	CA_LogImportSecret
LUNA_LOG_EXPORT_SECRET	CA_LogExportSecret

CHAPTER 2: Backup and Restore

This chapter describes how to backup and restore the contents of your HSMs. It contains the following sections:

> "Backup & Restore Overview" below
> "Backup HSM Installation, Storage, and Maintenance" on page 44
> "Backup your HSM" on page 50
> "Remote Backup Service" on page 51
> "Restore your HSM Partition Locally" on page 74
> "Restore your HSM Partition from Token" on page 74
> "Troubleshooting and Frequently Asked Questions" on page 77

Backup & Restore Overview

HSM Partition backup securely clones partition objects from a named HSM Partition, to a SafeNet Luna Backup HSM (which is used whether you back up remotely or locally). This allows you to safely and securely preserve important keys, certificates, etc., away from the SafeNet appliance. It also allows you to restore the backup device's contents onto more than one HSM Partition, if you wish to have multiple partitions with identical contents.

HSM Partition backup command with the **add** option is a non-destructive process, where the contents of your HSM partition are copied to a matching partition on SafeNet Luna Backup HSM, adding new/changed objects to any that already exist on (that partition of) the backup device.

HSM Partition backup with the **replace** option is a destructive process (destructive to any material that might already exist on the target Backup partition - it does not affect objects on the partition that are being backed-up).

Backup for SafeNet Luna HSM 5 uses SafeNet Luna Backup HSM to backup and restore individual partitions.

The Backup device is a separately powered unit that can connect to the primary HSM in one of two ways:

> Locally, using direct connection at the host
> Remotely, via USB connection to a backup workstation with secure network connection to SafeNet Luna HSM's host

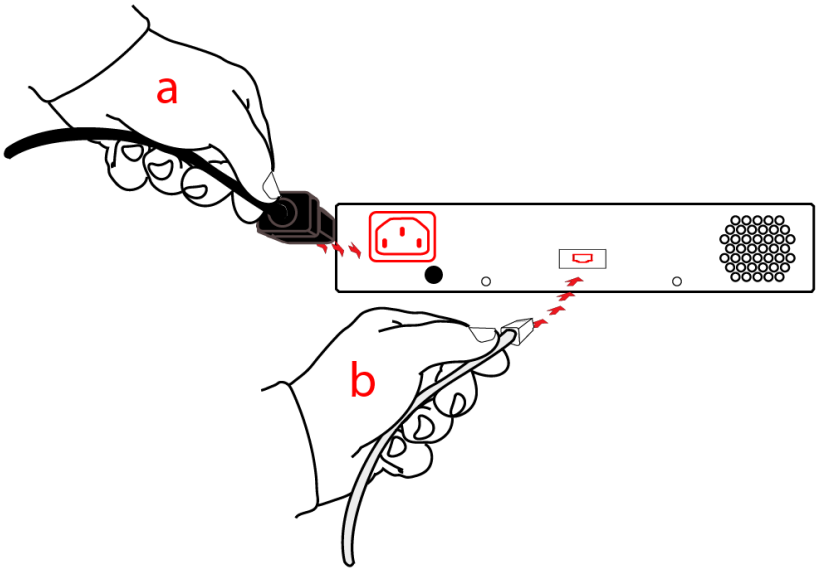
The backup operation looks a lot like the restore operation, because they are basically the same event, merely in different directions.

Connect

For local backup, connect SafeNet Luna Backup HSM to a power source, and via USB cable to the host's USB port.

For remote backup, connect SafeNet Luna Backup HSM to a power source, and via USB cable to a USB port on your computer.

In both cases, the cable attaches to the port on the back panel of SafeNet Luna Backup HSM, which requires a mini-USB at that end of the cable (similar cable as used to connect computers to cameras, cellphones, etc.)



For PED-authenticated HSMs

At the front panel, connect the SafeNet PED, using the supplied cable between the micro-D subminiature (MDSM) connector on top of the PED, and the matching MDSM connector on the front panel of SafeNet Luna Backup HSM (the connector labeled "PED").

Source and Target - full or partial

Issue the **partition backup** command.

Identify the partition to be backed up (source), and the partition that will be created (or added to) on the Backup HSM - the Token Partition Name.

Specify whether to **add** only unique objects (objects that have not previously been saved onto the target partition), or to completely **replace** the target partition (overwrite it).

In lunacm:> on a workstation, the command is:

```
lunacm:> partition backup backup -slot <slot> -pas <password> -par <backup partition>
```

This assumes that the target partition already exists with the appropriate domain.

Domain

If the target partition exists on the Backup HSM, then it must already share its partition domain with the source partition.

If the target partition is being created, then it takes the domain of the source partition.

Multiple partitions, with different domains, can exist on a single SafeNet Luna Backup HSM.

As with backup operations, restore operations can take place only where the source and target partitions have the same domain.

> Full/replace backup or restore creates a new target partition with the same domain as the source partition.

> Partial (additive/incremental) backup or restore requires the existing source and target partitions to have the same domain before the operation can start.

No cross-domain copying (backup or restore) is possible - there is no way to "mix and match" objects from different domains.

Replace or Append

If a matching target partition exists and the source partition is being incrementally backed up - choosing the **add** option in the command - then the target partition is not erased. Only source objects with unique IDs are copied to the target (backup) partition, adding them to the objects already there.

If a matching target partition exists and the source partition is being fully backed up - choosing the **replace** option in the command - then the existing partition is erased and a new one created.

PED or Password

SafeNet Luna Backup HSM creates a partition with matching authentication type to the SafeNet Luna PCIe HSM partition that is being backed up.

That does not work in the opposite direction, however. SafeNet Remote Backup Device can restore a partition (or contents of a partition) only to a SafeNet Luna Network HSM of matching authentication type.

You cannot mix partition authentication types on one backup device. That is, if you have a PED-authenticated HSM and a password-authenticated HSM, you require two SafeNet Luna Backup HSMs. Normally this is not a concern because a given installation is likely to employ all SafeNet Luna HSMs of the same authentication type in order to have a backup of each HSM's partitions. There is no possibility of backing up data from a higher-security device (Trusted Path, PED-authenticated, FIPS-3) onto a lower-security device (Password protected, FIPS-2).

However, for HSMs of the same authentication type, you could backup (or restore) partitions from different HSMs onto a single SafeNet Luna Backup HSM, as long as there is sufficient room. Given that the type matches, the authentication (domain) is handled at the partition level.

Remote Backup and Restore

Remote backup and restore follow the rules for local backup and restore, with some additional considerations.

When used in Remote mode, SafeNet Luna Backup HSM is connected via USB to a workstation computer that can be the same host that contains the primary HSM, or can be physically distant.

As of SafeNet Luna HSM 5.2 release, it is convenient to use a single Luna PED (Remote) for PED interaction with both local and remote HSMs.

About HSM Backup - Local and Remote

In many cases, it is sound practice to back up the contents of your SafeNet Luna PCIe HSM, in particular the contents of HSM partitions.

If the important objects are static, then a single backup is sufficient. If important objects change frequently, or if it is important to be able to revert to an identifiable date/time/condition/content, then regular backups are a necessity.

The Backup HSM

SafeNet Luna HSM 5.x backup is performed with the SafeNet Luna Backup HSM. Note that the word "Remote" in that product name merely denotes a capability. The SafeNet Luna Backup HSM also works fine as the local backup device for SafeNet Luna HSM, and is the only device supported for both local or remote backup of SafeNet Luna PCIe HSM.

The options for backup of primary/source SafeNet Luna HSMs are:

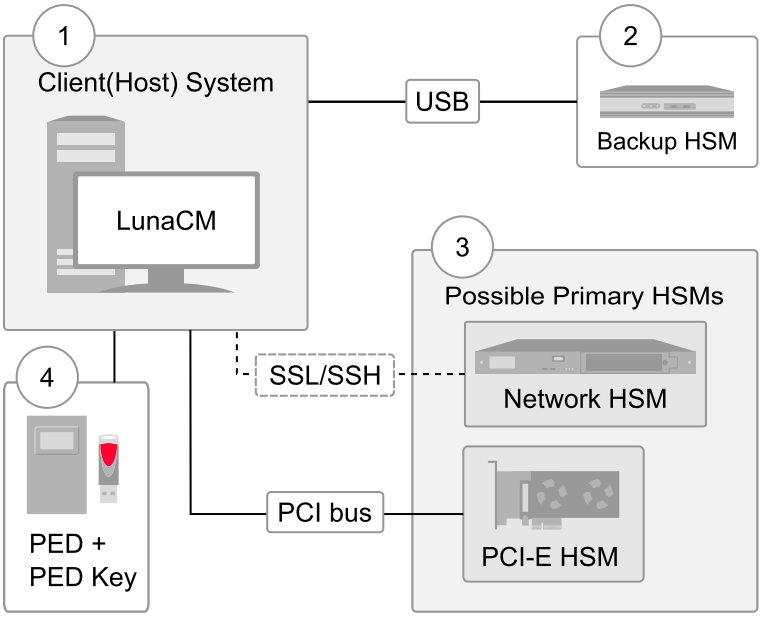
- **Local backup of any SafeNet Luna HSM**, where all components are co-located. This is a possible scenario with all SafeNet Luna HSMs, but is more likely with direct-connect, local-to-the-client HSMs such as SafeNet Luna PCIe HSM. It is unlikely for SafeNet Luna Network HSM, simply because SafeNet Luna Network HSM normally resides in a server rack, distant from its administrators.
- **Local backup of SafeNet Luna PCIe HSM**, where SafeNet Luna Network HSM is located remotely from a computer that has the SafeNet Luna Backup HSM. This is one of the likely scenarios with SafeNet Luna Network HSM, but requires that the administrator performing backup must have client authentication access to all SafeNet Luna Network HSM partitions.
- **Remote backup of any SafeNet HSM**, where the SafeNet Luna HSM is located remotely from the computer that has the SafeNet Luna Backup HSM. This scenario requires that the administrator of the SafeNet Luna Backup HSM's host computer connects (via SSH or RDP) to the clients of each HSM partition that is to be backed up. The client performs the backup (or restore) under remote direction.

In Local mode, you connect directly to SafeNet Luna PCIe HSM via USB. That is, local backup is local to the HSM appliance being backed-up, not necessarily local to the administrator who is directing the process, who might be far away.

For remote backup, you connect (again via USB) to a computer running vtl and the driver for the device. Backup and restore are then performed over the secure network connection. For PED-authenticated SafeNet Luna PCIe HSM, you must have a copy of the appropriate red (domain) PED keys, from the SafeNet Luna PCIe HSM, to use with the Backup HSM, in order to perform the copy /cloning (backup and restore) operation between the HSMs.

Local Backup of co-located HSMs

The following diagram depicts the elements and connections of the local backup (and restore) operation, where everything is in one room.



1	LunaCM on Client (Host) System sees the primary and backup slots and controls the backup/restore operation
2	Backup HSM is a slot visible to "Client (Host) System" when Client (Host) System runs LunaCM
3	Primary HSMs are slots visible to "Client (Host) System" when Client (Host) System runs LunaCM
4	Every slot on the backup must have same domain (red PED key) as matching slot on the primary HSMs

For SafeNet Luna Network HSM, the above would be a minority scenario.

The other two backup and restore options:

> Local backup of a distant SafeNet Luna Network HSM

> Remote backup of any SafeNet Luna HSM

... require that PED operations be performed remotely. For that reason, HSMs must be prepared (locally) in advance by having orange Remote PED keys created and matched with each HSM.

Local Backup of a Distant SafeNet HSM

This applies only to SafeNet Luna Network HSM, and is not an option for SafeNet Luna PCIe HSM.

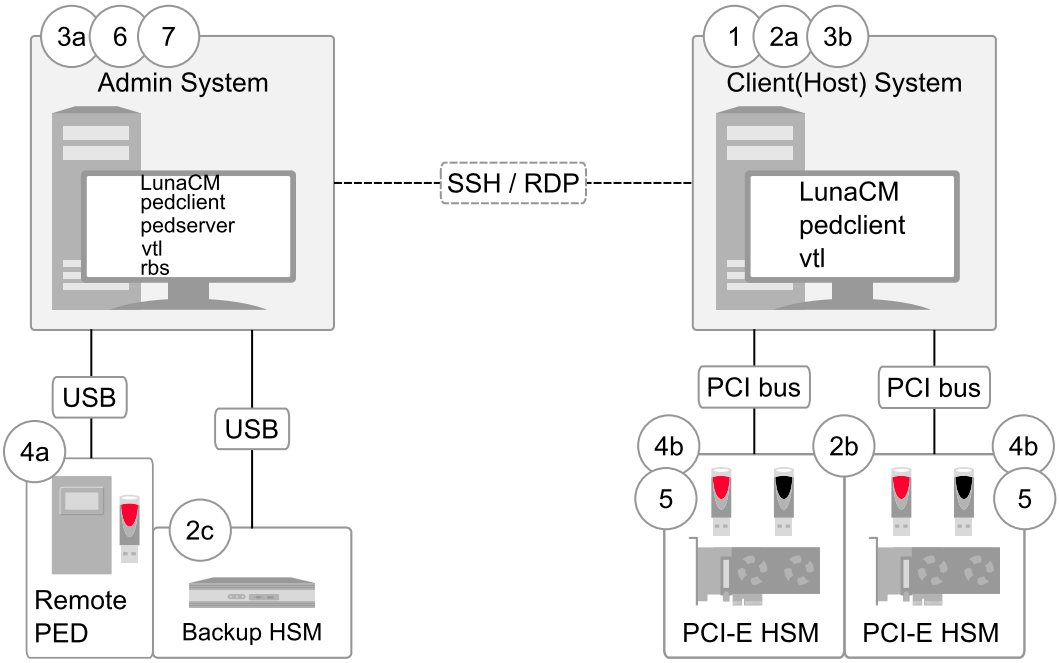
Preparing (configuring) for Remote Backup with Remote PED

While it is standard to remotely administer SafeNet Luna Network HSM, you can also remotely administer an HSM (SafeNet Luna PCIe HSM embedded in a distant host computer by means of an SSH session or an RDP (Remote Desktop Protocol) session. You could administer several such HSMs from a central location, including performing backup and restore operations with a SafeNet Luna Backup HSM connected to your Admin computer (perhaps a laptop).

For PED-authenticated HSMs, this operation requires a PED connection to each primary SafeNet Luna HSM and someone to insert PED keys and press buttons on the PED keypad, which implies Remote PED and Remote Backup. Once the HSM has been matched to an orange Remote PED key, all future authentications can be performed with Remote PED, and the HSM can safely be deployed to its distant location.

Remote Backup

In the following diagram, the preparation (above) has been done, and suitable orange and red PED keys have the appropriate secrets imprinted, to allow Remote PED connection and Remote (or Local) Backup (cloning) respectively.



This scenario is applicable to both SafeNet Luna PCIe HSM and SafeNet Enterprise HSMs with slight differences in handling.

1	LunaCM is on both the Client (Host) System and the Admin System, but is run on Client (Host) System to launch and manage the backup and restore activity.
2	LunaCM on "Client (Host) System" (2a) sees the primary (2b) and backup (2c) slots and controls the backup/restore.
3	> PedClient is needed on both the Client (Host) System and the Admin System > PedClient is needed on any host that must reach out to a pedserver instance and a Remote PED > PedClient instances can also communicate with each other to facilitate RBS
4	Every slot on the backup (4a) must have the same domain (red KED Key) as the matching slot on the primary HSMs (4b).
5	Every primary HSM slot (partition) that is to be backed up or restored must be in login or activated state (black PED keys -(5)), so that the Client (Host) System can access it with lunacm:> backup or restore commands.

6	PedServer must reside (and run, waiting for calls) on any computer connected to a Remote PED.
7	RBS is required on the computer connected to the SafeNet Luna Backup HSM. RBS is not needed on any other computer in the scenario.

As noted previously, the orange PED keys (Remote PED keys or RPK) contain a Remote PED Vector (RPV) that matches the RPV inside the SafeNet Luna HSM. It is the presence of that RPV at both ends that allows the connection to be made between the HSM and the Remote PED.

About HSM Backup - Local and Remote

In many cases, it is sound practice to back up the contents of your SafeNet Luna Network HSM, in particular the contents of HSM partitions.

If the important objects are static, then a single backup is sufficient. If important objects change frequently, or if it is important to be able to revert to an identifiable date/time/condition/content, then regular backups are a necessity.

The Backup HSM

SafeNet Luna HSM 5.x backup is performed with the SafeNet Luna Backup HSM. Note that the word "Remote" in that product name merely denotes a capability. The SafeNet Luna Backup HSM also works fine as the local backup device for SafeNet Luna HSM, and is the only device supported for both local or remote backup of SafeNet Luna Network HSM.

The options for backup of primary/source SafeNet Luna HSMs are:

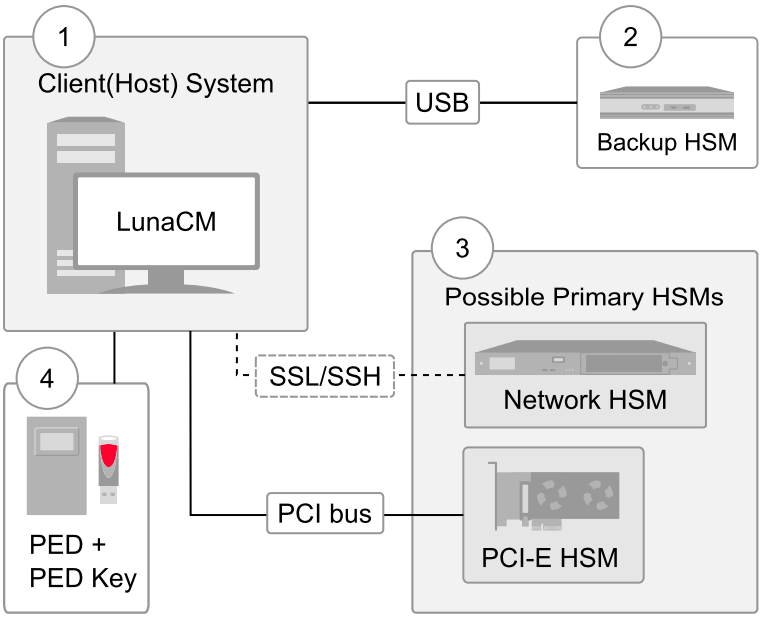
- **Local backup of any SafeNet Luna HSM**, where all components are co-located. This is a possible scenario with all SafeNet Luna HSMs, but is more likely with direct-connect, local-to-the-client HSMs such as SafeNet Luna PCIe HSM. It is unlikely for SafeNet Luna Network HSM, simply because SafeNet Luna Network HSM normally resides in a server rack, distant from its administrators.
- **Local backup of SafeNet Luna Network HSM**, where SafeNet Luna Network HSM is located remotely from a computer that has the SafeNet Luna Backup HSM. This is one of the likely scenarios with SafeNet Luna Network HSM, but requires that the administrator performing backup must have client authentication access to all SafeNet Luna Network HSM partitions.
- **Remote backup of any SafeNet HSM**, where the SafeNet Luna HSM is located remotely from the computer that has the SafeNet Luna Backup HSM. This scenario requires that the administrator of the SafeNet Luna Backup HSM's host computer connects (via SSH or RDP) to the clients of each HSM partition that is to be backed up. The client performs the backup (or restore) under remote direction.

In Local mode, you connect directly to SafeNet Luna Network HSM via USB. That is, local backup is local to the HSM appliance being backed-up, not necessarily local to the administrator who is directing the process, who might be far away.

For remote backup, you connect (again via USB) to a computer running vtl and the driver for the device. Backup and restore are then performed over the secure network connection. For PED-authenticated SafeNet Luna Network HSM, you must have a copy of the appropriate red (domain) PED keys, from the SafeNet Luna Network HSM, to use with the Backup HSM, in order to perform the copy /cloning (backup and restore) operation between the HSMs.

Local Backup of co-located HSMs

The following diagram depicts the elements and connections of the local backup (and restore) operation, where everything is in one room.



1	LunaCM on Client (Host) System sees the primary and backup slots and controls the backup/restore operation
2	Backup HSM is a slot visible to "Client (Host) System" when Client (Host) System runs LunaCM
3	Primary HSMs are slots visible to "Client (Host) System" when Client (Host) System runs LunaCM
4	Every slot on the backup must have same domain (red PED key) as matching slot on the primary HSMs

For SafeNet Luna Network HSM, the above would be a minority scenario.

The other two backup and restore options:

> Local backup of a distant SafeNet Luna Network HSM

> Remote backup of any SafeNet Luna HSM

... require that PED operations be performed remotely. For that reason, HSMs must be prepared (locally) in advance by having orange Remote PED keys created and matched with each HSM.

Local Backup of a Distant SafeNet HSM

This applies only to SafeNet Luna Network HSM, and is not an option for SafeNet Luna PCIe HSM.

Preparing (configuring) for Remote Backup with Remote PED

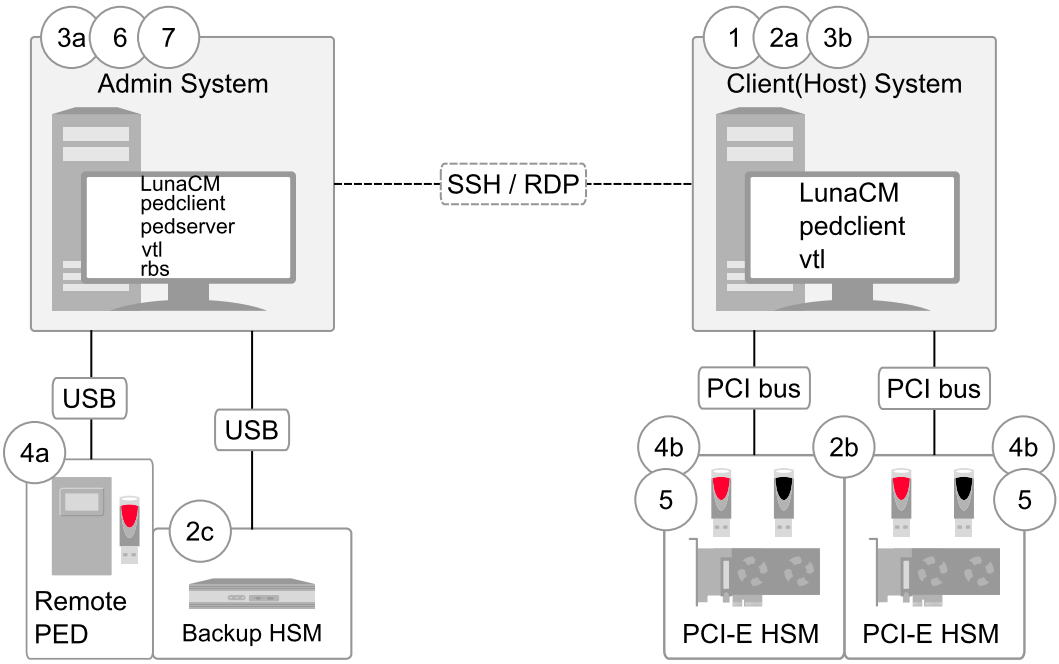
While it is standard to remotely administer SafeNet Luna Network HSM, you can also remotely administer an HSM (SafeNet Luna PCIe HSM embedded in a distant host computer by means of an SSH session or an RDP

(Remote Desktop Protocol) session. You could administer several such HSMs from a central location, including performing backup and restore operations with a SafeNet Luna Backup HSM connected to your Admin computer (perhaps a laptop).

For PED-authenticated HSMs, this operation requires a PED connection to each primary SafeNet Luna HSM and someone to insert PED keys and press buttons on the PED keypad, which implies Remote PED and Remote Backup. Once the HSM has been matched to an orange Remote PED key, all future authentications can be performed with Remote PED, and the HSM can safely be deployed to its distant location.

Remote Backup

In the following diagram, the preparation (above) has been done, and suitable orange and red PED keys have the appropriate secrets imprinted, to allow Remote PED connection and Remote (or Local) Backup (cloning) respectively.



This scenario is applicable to both SafeNet Luna PCIe HSM and SafeNet Enterprise HSMs with slight differences in handling.

1	LunaCM is on both the Client (Host) System and the Admin System, but is run on Client (Host) System to launch and manage the backup and restore activity.
2	LunaCM on "Client (Host) System" (2a) sees the primary (2b) and backup (2c) slots and controls the backup/restore.
3	> PedClient is needed on both the Client (Host) System and the Admin System > PedClient is needed on any host that must reach out to a pedserver instance and a Remote PED > PedClient instances can also communicate with each other to facilitate RBS

4	Every slot on the backup (4a) must have the same domain (red KED Key) as the matching slot on the primary HSMs (4b).
5	Every primary HSM slot (partition) that is to be backed up or restored must be in login or activated state (black PED keys -(5)), so that the Client (Host) System can access it with lunacm:> backup or restore commands.
6	PedServer must reside (and run, waiting for calls) on any computer connected to a Remote PED.
7	RBS is required on the computer connected to the SafeNet Luna Backup HSM. RBS is not needed on any other computer in the scenario.

As noted previously, the orange PED keys (Remote PED keys or RPK) contain a Remote PED Vector (RPV) that matches the RPV inside the SafeNet Luna HSM. It is the presence of that RPV at both ends that allows the connection to be made between the HSM and the Remote PED.

Backup HSM Installation, Storage, and Maintenance

This section describes how to install and maintain your SafeNet Luna Backup HSM (Backup HSM), and prepare it for storage. It contains the following sections:

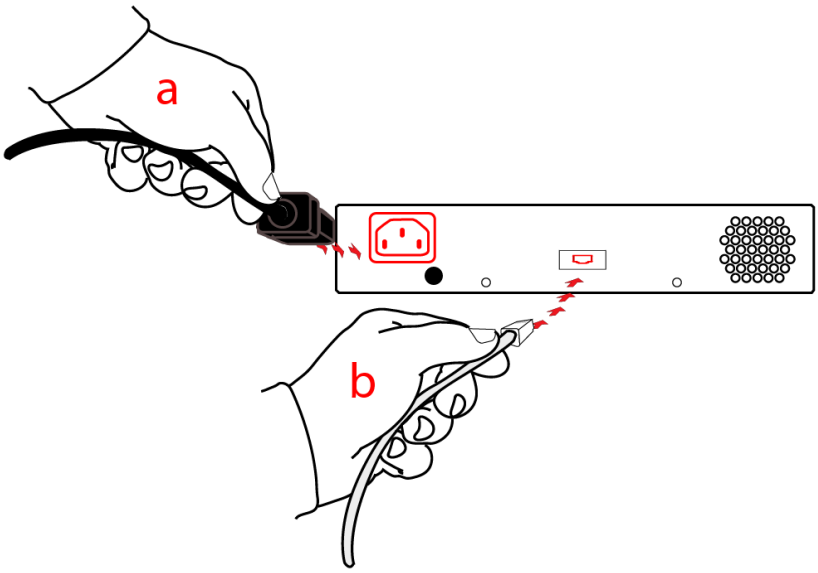
> "Connecting a Backup HSM" below
> "Disconnecting a Backup HSM" on page 46
> "Installing the Battery" on page 46
> "Backup HSM Storage and Maintenance" on page 48

Connecting a Backup HSM

For local backup, connect the Backup HSM to a power source, and via USB cable to the SafeNet Luna Network HSM USB port.

For remote backup, connect the Backup HSM to a power source, and via USB cable to a USB port on your computer.

In both cases, the cable attaches to the port on the back panel of the Backup HSM, which requires a mini-USB at that end of the cable (similar cable as used to connect computers to cameras, older cellphones, etc.).



PED-authenticated HSMs

At the front panel, connect the SafeNet PED, using the supplied cable between the micro-D subminiature (MDSM) receptacle on top of the PED, and the matching MDSM receptacle on the front panel of SafeNet Luna Backup HSM (the receptacle labeled "PED").

Disconnecting a Backup HSM

The Backup HSM is a USB device. It is not equipped with a power switch. There is no special procedure for disconnecting or shutting down a SafeNet Luna Backup HSM.

If the Backup HSM is used in remote configuration for SafeNet Luna PCIe HSM (connected to a workstation acting as backup server), then your only action is to do the usual dismount of a USB device (for the benefit of your workstation, not the Backup HSM - “It is now safe to disconnect your USB Device”). Linux and UNIX platforms have their equivalent unmount actions for USB. Then disconnect the cables.

If the Backup HSM is connected to SafeNet Luna Network HSM for local backup, you have no access to the SafeNet Luna Network HSM’s internal hardened kernel, so you cannot issue an un-mount instruction. Simply disconnect the cables and the system figures it out at either end. Both SafeNet Luna Network HSM and the Backup HSM accept this treatment very robustly.

Installing the Battery

The battery that powers the NVRAM and RTC in the SafeNet Luna Backup HSM is shipped uninstalled, in the packaging. This preserves the battery in case the unit spends a long time in transit or is stored in your warehouse as a spare. With the battery not inserted, the real-time clock and NVRAM are not depleting its charge to no purpose. If you are preparing a fresh-from-the-factory Backup HSM to place it into service, then you must install the battery before using the device.

1

Begin by removing the front face-plate. It is held in place by two spring clips. Grasp the face-plate firmly and pull to disengage the clips. Set the face-plate aside.

2

The battery compartment is to the right as you face the unit. The compartment cover is circular and has both raised dots and a recessed slot. Use finger-pressure against the dots, or the edge of a coin in the slot, to twist the battery compartment cover ¼ turn in a counter-clockwise direction. The cover should fall out easily.

3

Remove the battery from its packaging and align it at the opening of the SafeNet Luna USB HSM (or SafeNet Luna Backup HSM) battery compartment. The battery has a “+” sign near the end with the raised nub/bump. The flat end of the battery is the negative pole (-).

4	 Insert the battery, negative end first. The positive end (+) should protrude. The compartment is spring-loaded.
5	 Use the battery compartment cover to push the battery into the compartment, against the spring tension. Maintaining the pressure, align the two tabs on the inside of the cover with the two recessed indentations at the top and bottom of the compartment opening. With a little jiggling and a few trial pushes, the tabs should settle into those recesses, allowing the cover to seat flush with the front of the SafeNet Luna Backup HSM. Maintain the inward pressure and twist the cover ¼ turn clockwise to lock it in place. The battery is installed.
6	Replace the front-panel cover by aligning the clips with their respective posts and pushing until the clips grab the posts and the cover snaps in place.

Backup HSM Storage and Maintenance

The SafeNet Luna Backup HSM (for backing up and restoring HSM and partition contents) and the SafeNet Luna USB HSM (for PKI options) can be stored, with valuable contents, when not in use. The battery that powers the NVRAM and RTC in either device must be installed for use, but some questions commonly arise if

the device is to be stored for long periods.

Should I take the battery out when storing the HSM in a safe?
It is generally good practice to remove batteries when storing electronic devices, to preclude accidental damage from battery leakage. We use high-quality, industrial-grade batteries, that are unlikely to fail in a damaging fashion, but prudence suggests removing them, regardless. Also, if the unit is not in use, there is no need to maintain power to the RTC and NVRAM, so an externally stored battery will last longer.

If the battery is out, what happens?
If main power is not connected, and the battery dies, or is removed, then NVRAM and the system's Real Time Clock lose power. The working copy of the MTK is lost.

If the battery dies during operation, will I lose my key material? Will corruption occur?
The only key material that is lost is session objects (including working copies of stored keys) that are in use at the time. If the "originals" of those same objects are stored as HSM/partition objects, then they reside in non-volatile memory, and those are preserved.

There is no corruption of stored objects.

Where can I get a spare/replacement battery?
From any supplier that can match the specifications.

Technical Specifications:

- 3.6 V Primary lithium-thionyl chloride (Li-SOCl2)
- Fast voltage recovery after long term storage and/or usage
- Low self discharge rate
- 10 years shelf life
- Operating temperature range -55 ºC to +85 ºC
- U.L. Component Recognition, MH 12193

Storage Conditions:

Cells should be stored in a clean & dry area (less than 30 % Relative Humidity)

Temperature should not exceed +30 ºC

How do I know if the battery is dead or about to die? Can I check the status of the battery?
There is not a low battery indicator or other provision for checking status.

The battery discharge curve is such that the voltage remains constant until the very end of the battery life, at which point the discharge is extremely steep.

What must I do to recover function, and access to my key material, after battery removal/discharge?
Insert the battery, connect the HSM, power it up, and resume using it.

The MTK that was deleted by the tamper event (battery removal/discharge) is reconstituted from stored portions as soon as you log in. All your stored material is available for use.

Backup your HSM

Non-Partition Objects

The backup and restore operations are partition commands for HSM partition contents. There is no equivalent explicit backup or restore command for HSM Administrator / SO space objects - that is, objects that, for whatever reason, are not in the HSM partition User space. If you have objects stored in the HSM Adminstrator / SO space of your SafeNet Luna PCIe HSM, you can securely copy them to a locally connected HSM (such as a second SafeNet Luna PCIe HSM card in another slot in the host computer) with the **hsm clone** command.

To backup your HSM:

1. If the target HSM is SafeNet Luna USB HSM, then ensure that the HSM is connected to power and to the host computer's USB port. If the target is a SafeNet Luna PCIe HSM, ensure that it is installed in a nearby PCIe slot.
2. Start the LunaCM utility.
3. Login to the primary/source HSM as HSM Adminstrator / SO.
4. Run:

   ```
   hsm clone -objects <objecthandle> -slot <slot#oftargethsm> -password targetHSMAdminpassword
   ```

   for SafeNet Luna PCIe HSM with Password Authentication, or

   ```
   hsm clone -objects <objecthandle> -slot <slot#oftargethsm>
   ```

   for SafeNet Luna PCIe HSM with PED Authentication.

To restore the HSM contents, reverse the cloning direction.

HSM Partition Backup

Partition backup securely clones partition objects (not including HSM Adminstrator / SO objects that are contained on the HSM, but not within an HSM Partition) from the HSM Partition, to a SafeNet Luna Backup HSM.

The options are:

- Your SafeNet Luna Backup HSM is connected directly to your HSM's USB port, described below. Use this option when you have just one HSM installed in the host computer.
- Your SafeNet Luna Backup HSM is connected to an administrative computer that is located remotely from the host computer containing your HSM, which is covered separately on "Backup your HSM Partition Remotely" on page 61.

To backup your HSM partition:

To backup a partition on your SafeNet Luna PCIe HSM, to a directly connected SafeNet Luna Backup HSM, have the Backup HSM connected to the AC mains power and to your HSM.

1. Start the LunaCM utility.
2. Select the slot to be backed up (if you have more than one HSM installed in the host computer.

3. Login to the source partition as User.
4. At the LunaCM prompt, type :

   ```
   partition backup backup -slot direct -partition <partition-on-backup-hsm> -password <partition-challenge> -replace
   ```

   Note that the partition on the source HSM needs no identification, other than the slot, since there is just one partition per HSM. You identify the target partition on the target SafeNet Luna Backup HSM because the Backup HSM is capable of containing multiple partitions as a backup repository for multiple SafeNet Luna PCIe HSMs or as multiple backups (on different days) of the same source SafeNet Luna PCIe HSM. A simple identification scheme is to use the text label of the source HSM when naming the target partition.
5. The content of the selected partition is copied to the named partition on the directly connected SafeNet Luna Backup HSM.

Disconnecting SafeNet Luna Backup HSM or SafeNet Luna USB HSM

The SafeNet Luna Backup HSM or the SafeNet Luna USB HSM is a USB device. It is not equipped with a power switch.

There is no special procedure for disconnecting or shutting down a SafeNet Luna Backup HSM or SafeNet Luna USB HSM.

If the Backup HSM or the SafeNet Luna USB HSM is connected to a workstation or host computer, then your only action is to perform the usual Windows (or other) dismount of a USB device (for the benefit of your workstation, not the HSM - “It is now safe to disconnect your USB Device”). Linux and UNIX platforms have their equivalent un-mount actions for USB. Then disconnect the cables.

Remote Backup Service

RBS (Remote Backup Service) allows you to backup and restore between a SafeNet Luna Backup HSM and a hosted primary SafeNet Luna PCIe HSM, where the two are distant from each other, while separating the backup responsibility from HSM partition ownership. That is, the person responsible for administering the Backup workstation (with attached SafeNet Luna Backup HSM) does not have Owner/User authentication (black PED key) for the primary HSM's partition.

RBS is not a standalone feature. It is a service that facilitates certain scenarios when backing-up HSM partitions or restoring onto those partitions, using a backup HSM that is distant from the primary HSM and its host or client.

RBS is run on the computer that hosts the SafeNet Luna Backup HSM, only. Running RBS also requires running pedClient on that computer, as well as on the distant primary - the paired instances of pedClient form the communications link that makes RBS possible.

Examples of the primary HSM might be:

- A SafeNet Luna PCIe HSM in its host computer (where the PCIe HSM is a local slot when viewed by LunaCM on the host computer)
- A SafeNet Luna PCIe HSM partition, seen as a "local" slot in LunaCM on a computer that is a registered client of that SafeNet Luna PCIe HSM

See "Prepare RBS to Support Backup / Restore" on page 59.

Sample Setup and Deployment

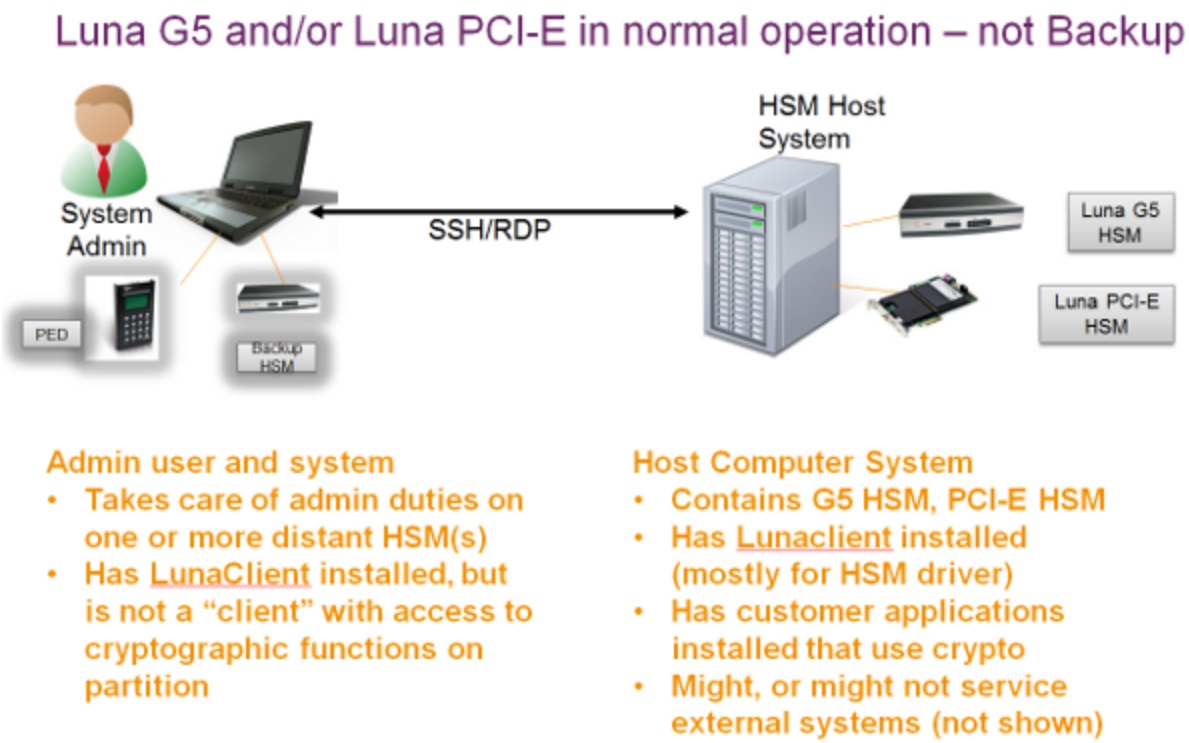
We will depict a sample deployment with SafeNet Luna USB HSM, the HSM that connects to a host computer via USB, and SafeNet Luna PCIe HSM, the HSM that is installed inside a host computer. Our choice is to consider the setup that the majority of customers seem to prefer:

1. A host computer with HSM residing in a secure room (server room, or other lock-up with restricted physical access)
2. An administrative workstation, often a laptop with both Remote PED and Remote Backup HSM equipment, communicating with the primary HSM via SSH or Remote Desktop Protocol sessions

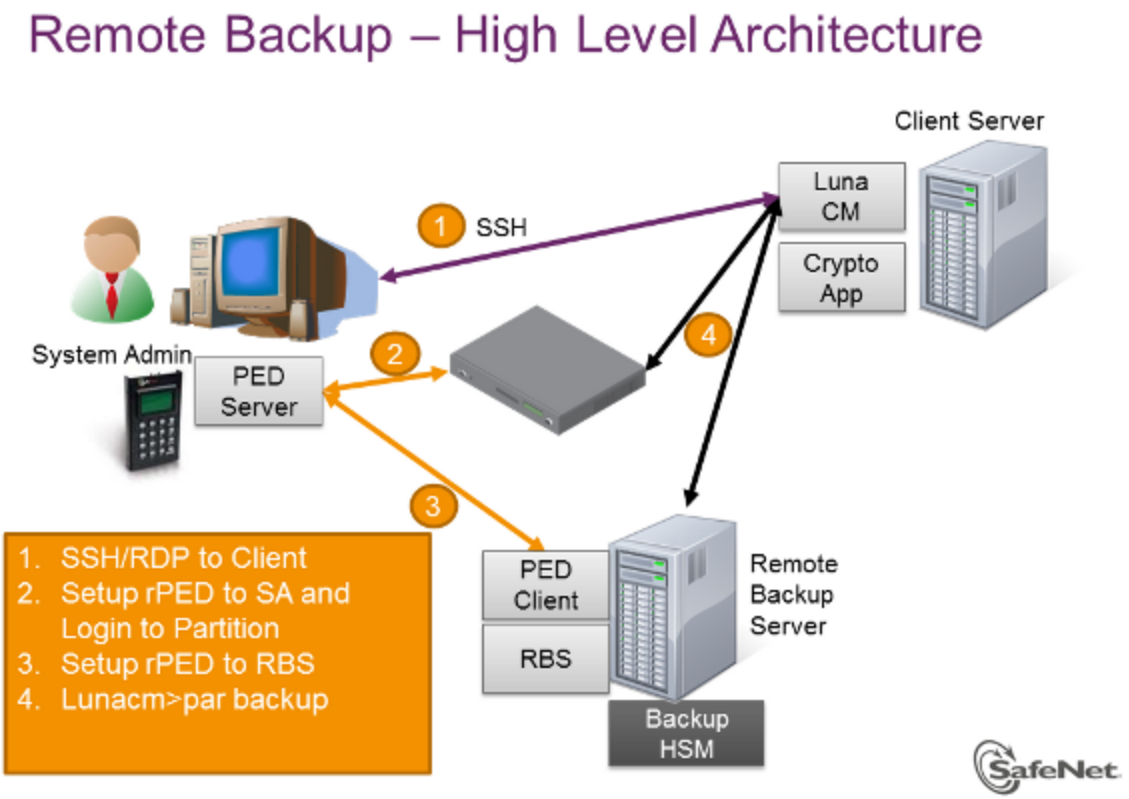
The HSM in the host takes care of cryptographic operations requested by client applications residing in the host computer.

The admin computer serves the HSM administrator who performs administrative and maintenance duties on behalf of the primary HSM on the host, including authentication for login and activation via Remote PED, and Remote Backup and Restore operations to/from the attached SafeNet Luna Backup HSM.

First, a look at the described setup in everyday operation, without considering Backup and Restore.



Here is the general case of Remote Backup, with the functions distributed on different computers.



Backup is controlled via the `lunacm:>` command line. As a system or security administrator, you choose which computer is to run `lunacm:>` to accomplish the backup/restore operation. The choice of approach comes down to the familiar trade-off between convenience and security.

The `lunacm:>` utility resides on the HSM's host computer and views the SafeNet Luna Backup HSM as a slot at an IP address (corresponding to an administrator's workstation). The administrator uses an SSH or RDP (Remote Desktop Protocol) session to connect to the primary HSM's host computer and to work that `lunacm:>` instance where it resides. That is, the administrator is not using `lunacm:>` on his own computer to run the backup operation. The backup administrator/operator is using `lunacm:>` on the computer that is directly attached to the primary HSM (the one with the partition being backed up, such as SafeNet PCIe HSM), or that is a client of a network-attached HSM partition (as in SafeNet Luna Network HSM).

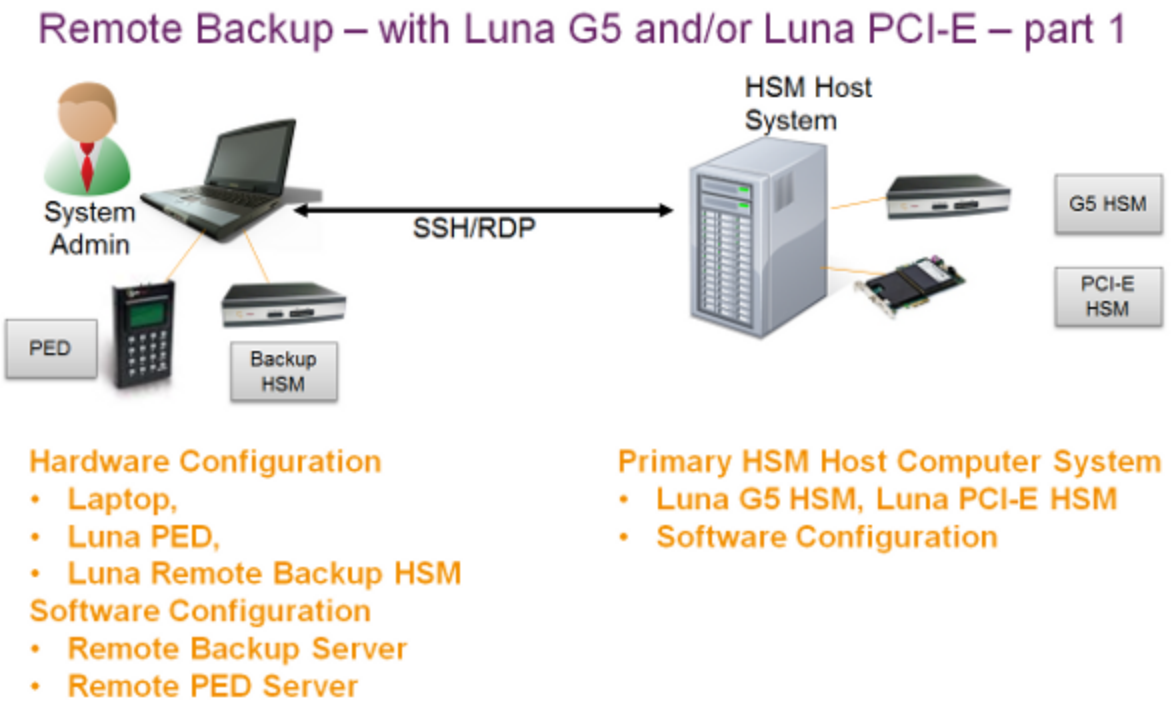
The `lunacm:>` session on the host computer views its embedded/attached HSMs as local slots. The `lunacm:>` session can see a distant SafeNet Luna Network HSM as a local slot if the HSM host computer has been made a client of a partition on that SafeNet Luna Network HSM (by a certificate exchange and registration.)

RBS is needed on the Remote Backup computer for this arrangement.

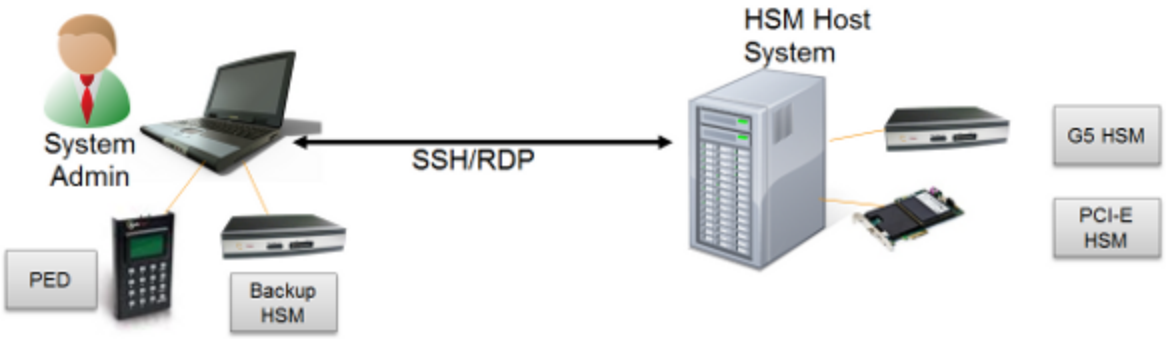
Other than that small difference of perspective, the Remote Backup function works identically for all primary SafeNet Luna HSMs. The drawback to this Remote Backup protocol is that one or more computers, distant from the Backup HSM must be used, as they must be clients of the SafeNet Luna HSM partitions. However, because established clients already have access to their registered partitions, the `lunacm:>` instance on each client computer can be employed to broker the Remote Backup operation, without exposing the partition access credentials to the operator of the Backup HSM computer. This maintains separation of roles.

The other option for an administrator wanting to back up a distant SafeNet Luna Network HSM partition is to make the computer with the Backup HSM a direct, registered client of the SafeNet Luna Network HSM. Then `lunacm:>` on that Backup HSM computer can see the distant SafeNet Luna Network HSM as a local slot. This is a local backup operation that does not use RBS, and does not require another computer in the process. The potential drawback is that the Backup HSM computer must have client access to every SafeNet Luna Network HSM partition that it backs up using Local Backup protocol. In some environments, this might be regarded as a security issue.

Next, a series depicting the setup and use of Remote Backup and Restore, assisted by Remote PED, where administrator, Remote PED, and Remote Backup are combined at a single laptop/workstation.



Remote Backup – with Luna G5 and/or Luna PCI-E – part 2

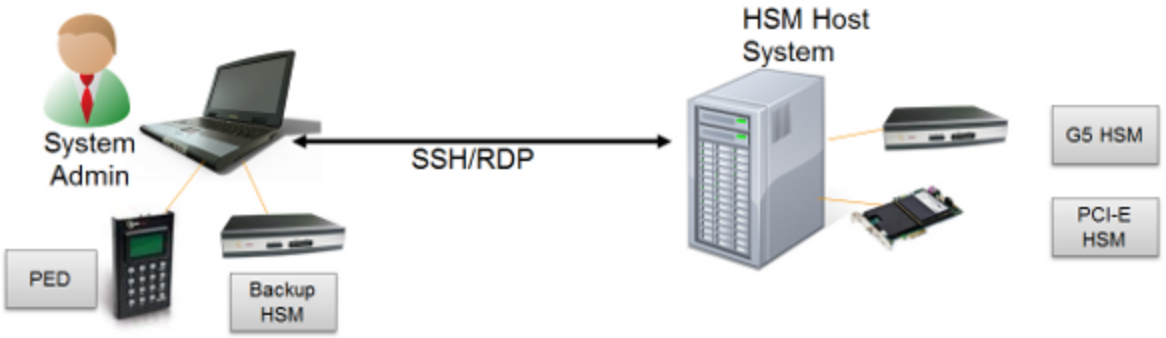


Set up RBS
- Configure
- Generate key
- Copy RBS certificate to HSM Host System
- Start RBS application

Set up PED Server
- Start PED Server application

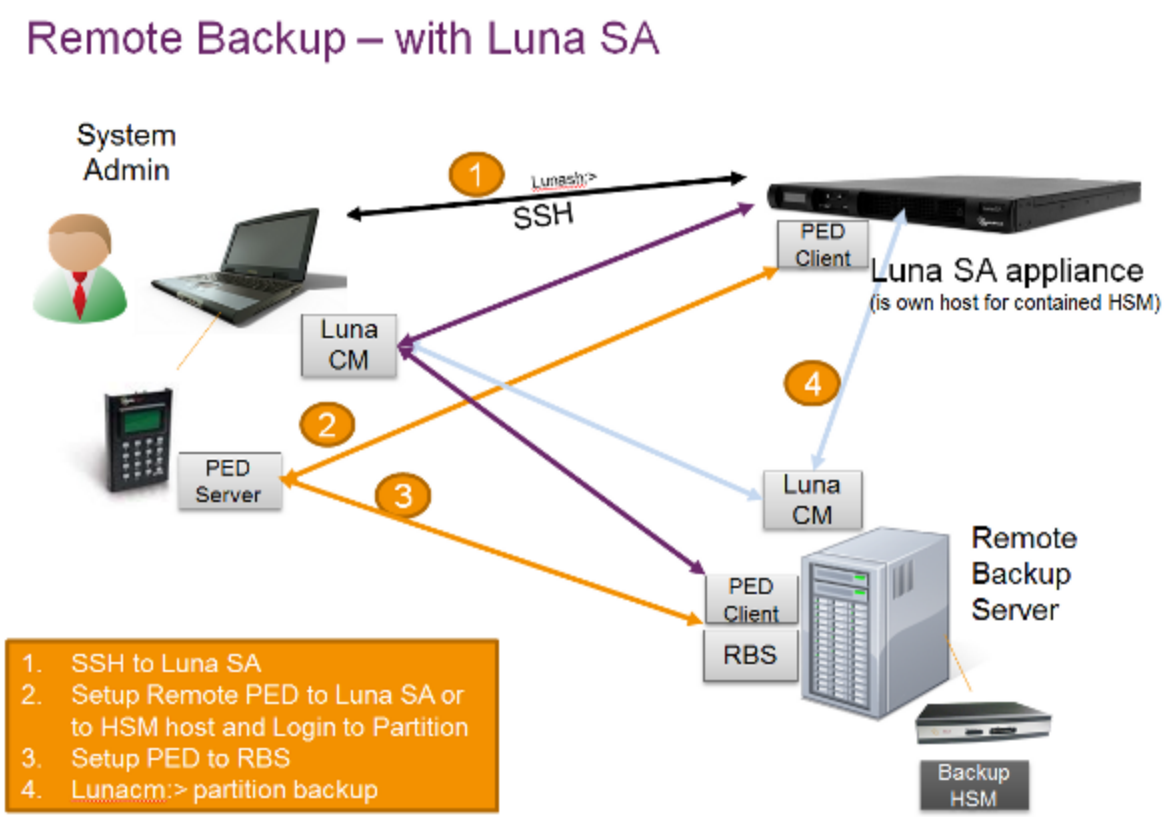
Set up HSM Host System
- Add RBS as server using "VTL"
- Start LunaCM

Remote Backup – with Luna G5 and/or Luna PCI-E – part 3

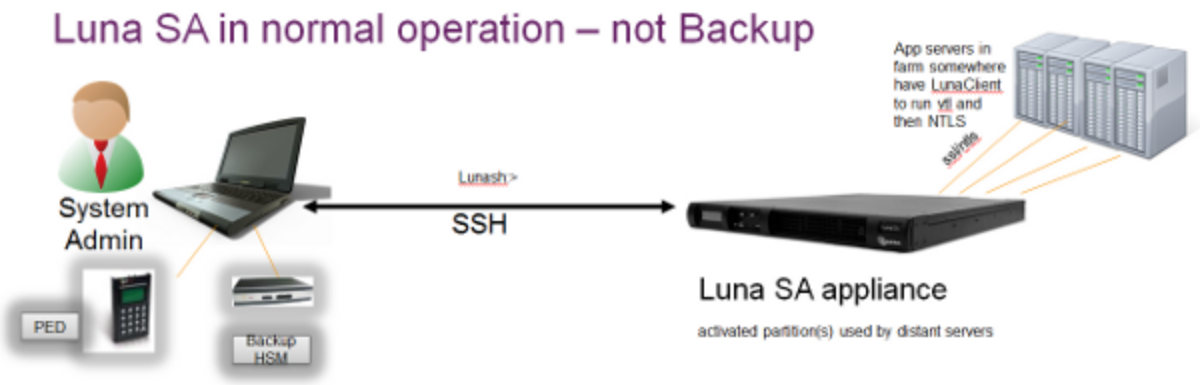


Perform Backup using LunaCM
- Set slot to HSM to be backed up
- Connect to the Remote PED
- Login the partition
- Disconnect Remote PED
- Connect Remote PED to remote backup slot
- Execute backup partition command
- Disconnect Remote PED from remote backup slot

Remote Backup with Remote PED for SafeNet Luna PCIe HSM, the overview.



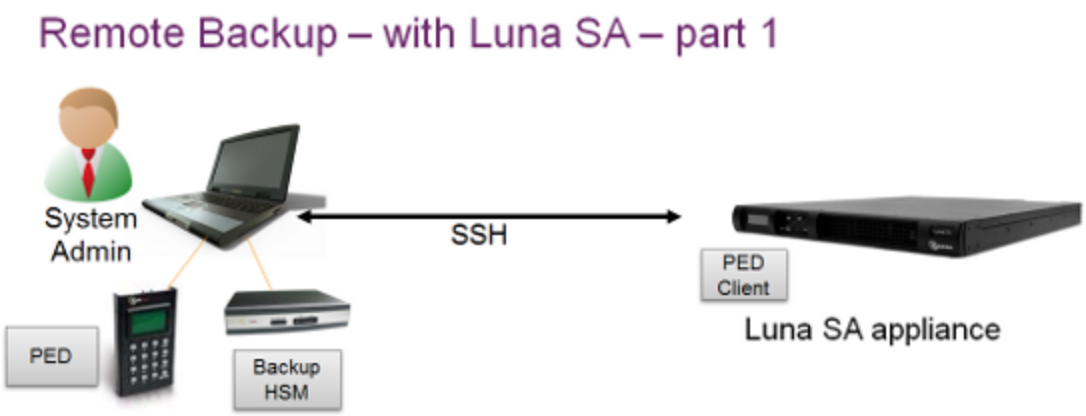
SafeNet Luna PCIe HSM as it would normally operate, serving clients, and being administered via lunash:> over SSH.



Admin user and system
- Takes care of admin duties on one or more distant HSM(s)
- Has LunaClient installed, but is not normally a "client" with access to cryptographic functions on partition
- Has lunacm and vtl

Luna SA appliance system
- Contains K6 (PCI-E) HSM
- Has Lunash:> installed
- Responds to application crypto calls from distant client servers, via NTLS
- Does not run any apps locally
- Administered by remote system admin (including backups)

Now, a sequence summarizing Remote Backup setup and use.

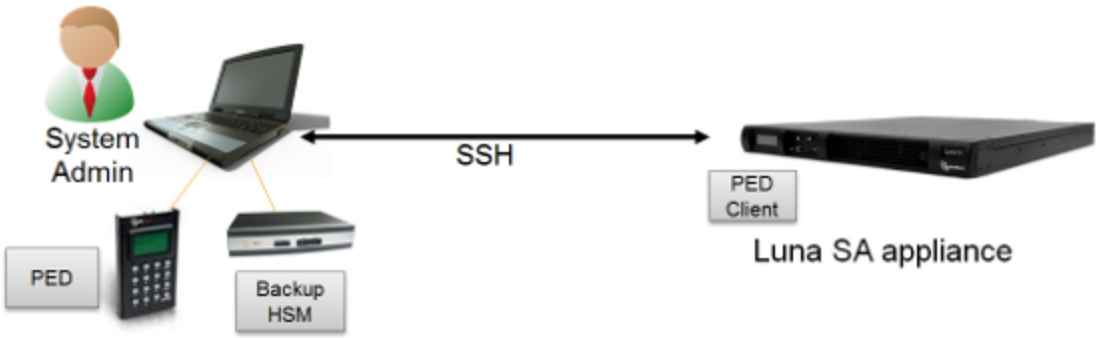


Hardware Configuration
- Laptop,
- Luna PED,
- Luna Remote Backup HSM

Software Configuration
- Remote Backup Server
- Remote PED Server

Primary HSM (with source partition)
- Luna SA appliance
- Software Configuration

Remote Backup – with Luna SA – part 2

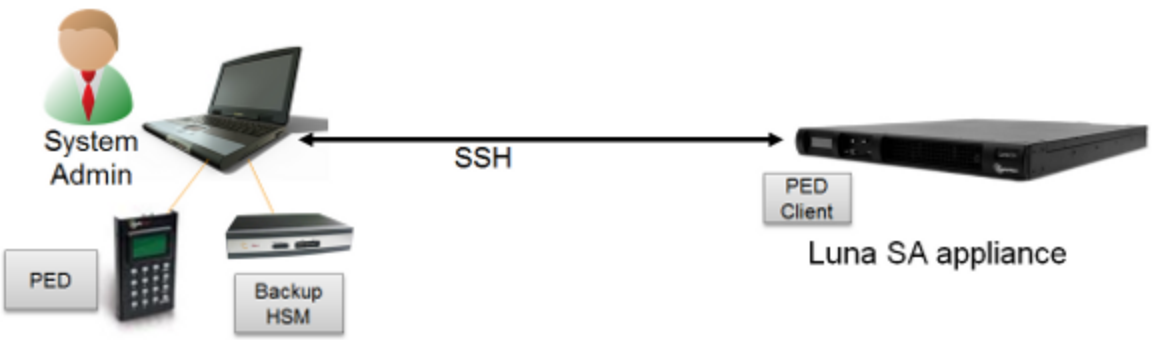


Set up RBS
- Configure
- Generate key
- Copy RBS certificate to HSM Host System
- Start RBS application

Set up PED Server
- Start PED Server application

Set up HSM appliance
- Add RBS as server using "VTL"
- Start LunaCM on admin/backup station

Remote Backup – with Luna SA – part 3



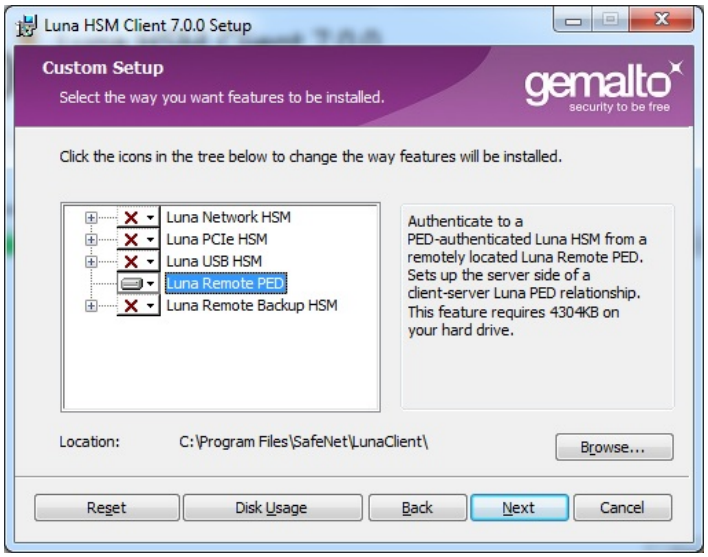
Perform Backup using LunaCM
- Set slot to HSM to be backed up
- Connect to the Remote PED
- Login the partition
- Disconnect Remote PED
- Connect Remote PED to remote backup slot
- Execute backup partition command
- Disconnect Remote PED from remote backup slot

Prepare RBS to Support Backup / Restore

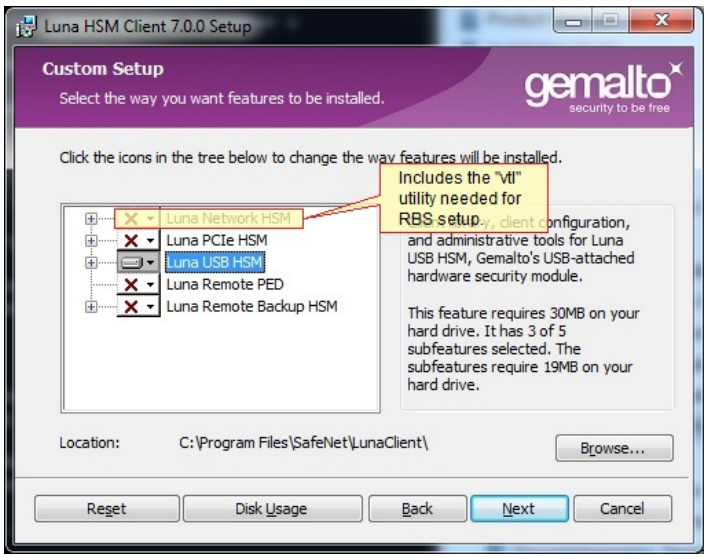
Remote Backup uses the Remote Backup Service (RBS), which must be installed and configured before you use it. RBS is a separate option at software installation time. You do not need it on all client/admin computers, but it doesn't hurt to have it installed.

To prepare for RBS:

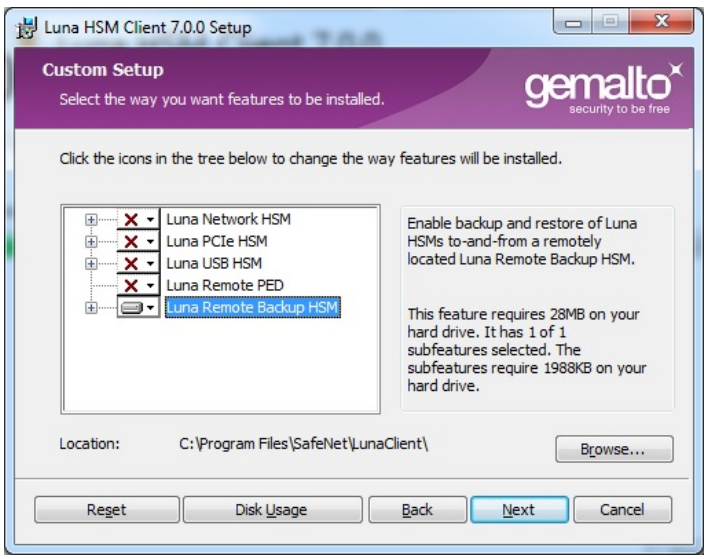
1. Install SafeNet Luna PCIe HSM Client software on the computer that will manage your primary HSM (could be the administrative client for SafeNet Luna PCIe HSM, or the host computer containing one or more SafeNet Luna PCIe HSMs, or connected to one or more SafeNet Luna USB HSMs). Probably you will want to include the Remote PED option.



   If the primary HSM is other than SafeNet Luna PCIe HSM, install the SafeNet Luna PCIe HSM option in addition to the SafeNet Luna USB HSM or SafeNet Luna PCIe HSM software, because the SafeNet Luna PCIe HSM client is the only one that includes the **vtl** utility, necessary for the certificate exchange that enables Remote Backup Service.



2. Install SafeNet Luna PCIe HSM Client software for the host computer connected to your Backup HSM. Select the Remote Backup option.



You could also choose to install the Remote PED option here. It depends on how you intend to separate the functions and, other than the space it occupies on your hard disk, it doesn't hurt to have any of the SafeNet Luna PCIe HSM Client options installed and available.

3. Run **rbs --genkey** to generate the server.pem to establish the Remote Backup Service between the Backup host and the host/client for the primary HSM. The location of the server.pem file can be found in the Chrystoki.conf /crystoki.ini file.
4. Run **rbs --config** to specify devices to support.
5. Run **rbs --daemon** to launch the rbs daemon (Linux and UNIX) or the rbs console application (on Windows, it closes after every use) .

6. Copy the certificate generated earlier (server.pem) to your primary HSM host computer or SafeNet Luna PCIe HSM appliance:

```
# scp root@192.20.9.253:/usr/safenet/lunaclient/rbs/server/server.pem .

root@192.20.9.253's password:     *********

server.pem                | 1 kB |   1.2 kB/s | ETA: 00:00:00 | 100%
```

7. Run **vtl** on the host computer (or appliance) to add the RBS server to the server list:

```
 vtl add -n 192.20.9.253 -c server.pem
New server 192.20.9.253 successfully added to server list.
 vtl list
Server: 192.20.9.82     HTL required: no
   Server: 192.20.9.253    HTL required: no
```

Now go to "Backup your HSM Partition Remotely" below.

The PEDClient is half of the PEDServer/PEDClient duo that enables Remote PED service.

However, PEDClient is also used in the communication component of Remote Backup Service. So, PEDClient should run on all the platforms that have HSMs - where a SafeNet Luna USB HSM or SafeNet Luna PCIe HSM is installed (PEDClient is already inside SafeNet Luna Network HSM 5.2 and newer...) - and also on any system with the RBS application.

The PEDServer is required only on a computer with the SafeNet Remote PED.

If you consolidate your HSM administration (including Remote PED) on the same computer with your SafeNet Remote Backup HSM, you would have both PEDClient and PEDServer installed there. We observe that a majority of customers combine administrative functions this way, on a laptop or a workstation that is used to administer one-or-many HSM hosts. The HSM host (with SafeNet Luna USB HSM or SafeNet Luna PCIe HSM) or the SafeNet Luna Network HSM appliance resides in a physically secure, possibly remote location, while the administrator works from a laptop in her/his office. Your security policy determines how you do it.

Backup your HSM Partition Remotely

The options to backup a partition on your SafeNet Luna PCIe HSM are:

- Local backup
- Remote backup

Local backup means that the SafeNet Remote Backup Device is co-located and physically connected to the SafeNet Luna PCIe HSM whose contents are to be backed up (that could be a SafeNet Luna PCIe HSM card inside a host computer, or a SafeNet Luna PCIe HSM appliance which is its own host for its internal HSM card).

In the case of SafeNet Luna PCIe HSM, you would most likely be using a laptop near the SafeNet Luna PCIe HSMM appliance to run your admin session (either by network SSH session or by a local serial connection), and would use locally connected Luna PEDs to provide the necessary authentication.

Remote backup means that the SafeNet Luna PCIe HSM in its host or appliance is at a remote location and you are working from a network connected computer where you open your SSH connection to the host (or SafeNet Luna PCIe HSM admin) shell, and you also have the SafeNet Luna Backup HSM connected to the computer, at least one SafeNet PED (which must be remote-capable), and the PED workstation software running.

Remote Backup Requirements

You will need:

Quantity	Description
1	SafeNet Luna PCIe HSM 5.2 or newer
1	Windows computer with SafeNet Luna PCIe HSM 5.2 (or newer) client software installed
1	SafeNet Luna Backup HSM
1	Set of PED keys imprinted for the source HSM and partitions
1	Luna PED 2 (Remote PED with f/w 2.5.0 or later)*
1	Power cable for Luna PED 2 (Remote)
2	USB to mini USB cable for Luna PED 2 (Remote) and SafeNet Luna Backup HSM

* The Luna PED that is connected to the Windows computer, in order to perform Remote PED operations with the distant SafeNet Luna PCIe HSM appliance, must be a Luna PED 2 (remote-capable version) and is used in Remote mode and in Local mode. You also have the option to connect a second SafeNet PED, which can be Remote capable or can be a local-only version, to the SafeNet Luna Backup HSM. This allows you to leave the Remote capable SafeNet PED connected to the workstation in Remote mode.

Assumptions

The following examples assume that you have set up RBS, as described in "Prepare RBS to Support Backup / Restore" on page 59.

- SafeNet Luna Backup HSM and your primary (source) SafeNet HSM are initialized with appropriate keys (blue SO and black Partition Crypto Officer/User PED keys, which can be the same for both devices, or can be different).
- Both devices must share the same domain or red PED key value.
- The workstation (Windows computer) has Remote PED and SafeNet Remote Backup software package installed including the appropriate driver.
- For SafeNet Luna PCIe HSM, NTLS is established between your workstation computer, acting as a SafeNet Luna PCIe HSM client, and the distant SafeNet Luna PCIe HSM - that is, the workstation is registered as a client with the partition.
- Remote PED session key (orange RPV key) has been created and associated with the distantSafeNet Luna PCIe HSM.

Before you begin setup of RBS:

1. Ensure that your Windows workstation has the PED USB driver (from the /USBDriver folder on the software CD) installed, and that the PEDServer.exe file (the executable program file that makes Remote PED operation possible) has been copied to a convenient directory on your hard disk.
2. Connect all of the components as follows:

From	Using	To
Workstation	USB	Remote PED (Luna PED IIr in Remote mode)
DC power receptacle on Remote PED	PED Power Supply	mains AC power (wall socket)
Workstation	USB	SafeNet Luna Backup HSM
SafeNet Luna Backup HSM	Power Cord	mains AC power (wall socket)
SafeNet Luna Backup HSM	Micro-D to Micro-D (local PED) cable	Luna PED (can be a separate local-or-Remote PED, or can be your single Remote PED set to operate in local mode for the local connection)

3. At the Remote Luna PED (Luna PED IIr connected to the USB port of the workstation:
   a. Press **<** on the PED keypad to exit Local mode
   b. Press **7** to enter Remote PED mode.
4. Start remote PED service on the administrative workstation (Windows) computer in a Command Prompt (DOS) window, change directory to the location of the PEDServer.exe file and run that file:

```
C:\>cd \Program Files\LunaClient
C:\Program Files\LunaClient>PEDServer -mode start
```

5. Open an administrative connection (SSH) to the distant SafeNet Luna PCIe HSM (for SafeNet Luna PCIe HSM appliance, log in as "admin". For another HSM host, log in with the appropriate ID. Start the PED Client (the Remote PED enabling process on the appliance):

```
            lunash:> hsm ped connect -ip {ip_workStation} -port 1503
or
lunacm:> hsm ped connect -ip {ip_workStation} -port 1503
```

   Insert the orange RPV PED key that matches the RPV of the distant SafeNet Luna PCIe HSM.
   The Remote PED Client in the SafeNet Luna PCIe HSM appliance or in the SafeNet Luna PCIe HSM or SafeNet Luna USB HSM host establishes a connection with the listening PED Server on your workstation.
6. Proceed to the Backup and Restore examples, below.

- "RBS Remote Backup with Single Remote PED on Windows " on the next page

- "Restore to a SafeNet Luna PCIe HSM Slot" on page 68

RBS Remote Backup with Single Remote PED on Windows

Just to indicate the versatility, this example uses a Windows 2012 64-bit client. PED Server is in Windows XP, SafeNet Luna Backup HSM is connected to Linux centOS 5.7.

This example shows a slot on a SafeNet Luna PCIe HSM being backed up. The same commands and sequences work for a SafeNet Luna PCIe HSM on a host computer. Just choose the desired HSM slot.

Backup from a SafeNet Luna PCIe HSM slot

This example assumes that you have already "Prepare RBS to Support Backup / Restore" on page 59.

That is, briefly:

> You have SafeNet Luna PCIe HSM Client software installed for your primary HSM (source of objects to be backed up).

> You have SafeNet Luna PCIe HSM Client software installed with the RBS option on the host computer connected to your Backup HSM.

> You have run RBS to generate private key/certificate, run RBS again to configure (select device(s) to support), run RBS again to launch the daemon (Linux/UNIX) or the service (Windows).

> You have copied the certificate (server.pem) to your primary HSM host computer (or SafeNet Luna PCIe HSM appliance).

> You have run **vtl** on the host computer (or appliance) to add the RBS server to the server list.

To backup from a SafeNet Luna PCIe HSM HSM slot:

1. Start the LunaCM utility (in Windows, it resides at C:\Program Files\SafeNet\LunaClient - in Linux/UNIX, it resides at /usr/safenet/lunaclient/bin):

```
C:\Program Files\SafeNet\LunaClient>lunacm.exe

LunaCM V7.0.0 - Copyright (c) 2006-2016 Gemalto, Inc.


        Available HSM's:

        Slot Id ->              1
        HSM Label ->            SA82_P1
        HSM Serial Number ->    500409014
        HSM Model ->            LunaSA
        HSM Firmware Version -> 6.10.1
        HSM Configuration ->    SafeNet Luna PCIe HSM HSM Slot (PED) Signing With Cloning Mode
        HSM Status ->           OK

        Slot Id ->              2
        HSM Label ->            G5PKI
        HSM Serial Number ->    701968008
        HSM Model ->            LunaSA
        HSM Firmware Version -> 6.10.1
        HSM Configuration ->     SafeNet Luna PCIe HSM Slot (PED) Signing With Cloning Mode
        HSM Status ->           OK

        Slot Id ->              3
```

```
        HSM Label ->            G5backup
        HSM Serial Number ->    700101
        HSM Model ->            G5Backup
        HSM Firmware Version -> 6.10.1
        HSM Configuration ->    Remote Backup HSM (PED) Backup Device
        HSM Status ->           OK

        Slot Id ->              4
        Tunnel Slot Id ->       6
        HSM Label ->            PCI422
        HSM Serial Number ->    500422
        HSM Model ->            K6 Base
        HSM Firmware Version -> 6.2.1
        HSM Configuration ->    Luna PCI (PED) Signing With Cloning Mode
        HSM Status ->           OK

        Slot Id ->              5
        Tunnel Slot Id ->       7
        HSM Label ->            K6_328
        HSM Serial Number ->    155328
        HSM Model ->            K6 Base
        HSM Firmware Version -> 6.10.1
        HSM Configuration ->    Luna PCI (PED) Signing With Cloning Mode
        HSM Status ->           OK

        Slot Id ->              8
        HSM Label ->            G5180
        HSM Serial Number ->    700180
        HSM Model ->            G5Base
        HSM Firmware Version -> 6.10.1
        HSM Configuration ->    SafeNet Luna USB HSM (PED) Signing With Cloning Mode
        HSM Status ->           OK


        Current Slot Id: 1
```

2. If the current slot is not the slot that you wish to backup, use the lunacm:> **slot set** command:

```
lunacm:> slot set slot 1

        Current Slot Id: 1     ( SafeNet Luna PCIe HSM Slot 6.10.1 (PED) Signing With Cloning
Mode)


Command Result : No Error
```

3. Establish that the HSM is listening for a Luna PED at the correct location (local or remote). In this example, we want the HSM to use a Luna PED that is not directly connected to the HSM - a Remote PED, at a specific location. The pedserver must already have been set up at that host.

```
lunacm:>ped get

        HSM slot 1 listening to local PED (PED id=0).

Command Result : No Error

lunacm:> ped connect ip 192.20.10.190
```

```
Command Result : No Error

lunacm:> ped get

        HSM slot 1 listening to remote PED (PED id=100).

Command Result : No Error
```

4. [Skip this step if your source partition is Activated]
   Log into the partition (this takes place at the currently selected slot). This step is needed only if the partition you are about to backup is not already in Activated state.

```
lunacm:> par login

        Option -password was not supplied.  It is required.

        Enter the password: *******

        User is activated, PED is not required.

Command Result : No Error
```

5. Disconnect the PED connection from your source HSM (slot 1 in this example), and connect to the SafeNet Luna Backup HSM (slot 3 in this example).

```
lunacm:> ped disconnect

        Are you sure you wish to disconnect the remote ped?

        Type 'proceed' to continue, or 'quit' to quit now -> proceed

Command Result : No Error

lunacm:> ped connect ip 192.20.10.190 -slot 3

Command Result : No Error

lunacm:> ped get -slot 3

        HSM slot 3 listening to remote PED (PED id=100).

Command Result : No Error
```

6. Perform the backup from the current slot (slot 1 in the example, see above) to the partition that you designate on the Backup HSM. Now that the Backup HSM is listening correctly for a PED, the target partition can be created, with PED action for the authentication.

```
lunacm:> partition backup backup -slot 3 -par SAbck1

        Logging in as the SO on slot 3.

        Please attend to the PED.

        Creating partition SAbck1 on slot 3.
```

```
Please attend to the PED.

Logging into the container SAbck1 on slot 3 as the user.

Please attend to the PED.

Creating Domain for the partition SAbck1 on slot 3.

Please attend to the PED.

Verifying that all objects can be backed up...

85 objects will be backed up.

Backing up objects...
Cloned object 99 to partition SAbck1 (new handle 19).
Cloned object 33 to partition SAbck1 (new handle 20).
Cloned object 108 to partition SAbck1 (new handle 23).
Cloned object 134 to partition SAbck1 (new handle 24).
Cloned object 83 to partition SAbck1 (new handle 25).
Cloned object 117 to partition SAbck1 (new handle 26).
Cloned object 126 to partition SAbck1 (new handle 27).
Cloned object 65 to partition SAbck1 (new handle 28).
Cloned object 140 to partition SAbck1 (new handle 29).
Cloned object 131 to partition SAbck1 (new handle 30).
Cloned object 94 to partition SAbck1 (new handle 31).
Cloned object 109 to partition SAbck1 (new handle 35).
Cloned object 66 to partition SAbck1 (new handle 36).
Cloned object 123 to partition SAbck1 (new handle 39).
Cloned object 74 to partition SAbck1 (new handle 40).
Cloned object 50 to partition SAbck1 (new handle 44).
Cloned object 43 to partition SAbck1 (new handle 45).
Cloned object 52 to partition SAbck1 (new handle 46).
Cloned object 124 to partition SAbck1 (new handle 47).
Cloned object 115 to partition SAbck1 (new handle 48).
Cloned object 98 to partition SAbck1 (new handle 49).
Cloned object 42 to partition SAbck1 (new handle 50).
Cloned object 48 to partition SAbck1 (new handle 51).
Cloned object 29 to partition SAbck1 (new handle 52).
Cloned object 54 to partition SAbck1 (new handle 53).
Cloned object 112 to partition SAbck1 (new handle 56).
Cloned object 69 to partition SAbck1 (new handle 57).
Cloned object 46 to partition SAbck1 (new handle 58).
Cloned object 116 to partition SAbck1 (new handle 59).
Cloned object 101 to partition SAbck1 (new handle 60).
Cloned object 122 to partition SAbck1 (new handle 61).
Cloned object 21 to partition SAbck1 (new handle 62).
Cloned object 45 to partition SAbck1 (new handle 63).
Cloned object 139 to partition SAbck1 (new handle 64).
Cloned object 127 to partition SAbck1 (new handle 65).
Cloned object 84 to partition SAbck1 (new handle 66).
Cloned object 30 to partition SAbck1 (new handle 70).
Cloned object 105 to partition SAbck1 (new handle 71).
Cloned object 132 to partition SAbck1 (new handle 72).
Cloned object 136 to partition SAbck1 (new handle 73).
Cloned object 28 to partition SAbck1 (new handle 74).
Cloned object 44 to partition SAbck1 (new handle 75).
Cloned object 26 to partition SAbck1 (new handle 76).
```

```
        Cloned object 120 to partition SAbck1 (new handle 77).
        Cloned object 104 to partition SAbck1 (new handle 78).
        Cloned object 137 to partition SAbck1 (new handle 79).
        Cloned object 61 to partition SAbck1 (new handle 80).
        Cloned object 110 to partition SAbck1 (new handle 81).
        Cloned object 125 to partition SAbck1 (new handle 82).
        Cloned object 129 to partition SAbck1 (new handle 83).
        Cloned object 53 to partition SAbck1 (new handle 84).
        Cloned object 130 to partition SAbck1 (new handle 85).
        Cloned object 73 to partition SAbck1 (new handle 86).
        Cloned object 41 to partition SAbck1 (new handle 87).
        Cloned object 135 to partition SAbck1 (new handle 88).
        Cloned object 114 to partition SAbck1 (new handle 89).
        Cloned object 22 to partition SAbck1 (new handle 90).
        Cloned object 57 to partition SAbck1 (new handle 91).
        Cloned object 79 to partition SAbck1 (new handle 92).
        Cloned object 121 to partition SAbck1 (new handle 96).
        Cloned object 34 to partition SAbck1 (new handle 97).
        Cloned object 103 to partition SAbck1 (new handle 98).
        Cloned object 89 to partition SAbck1 (new handle 99).
        Cloned object 128 to partition SAbck1 (new handle 103).
        Cloned object 119 to partition SAbck1 (new handle 104).
        Cloned object 107 to partition SAbck1 (new handle 105).
        Cloned object 118 to partition SAbck1 (new handle 106).
        Cloned object 111 to partition SAbck1 (new handle 107).
        Cloned object 133 to partition SAbck1 (new handle 108).
        Cloned object 138 to partition SAbck1 (new handle 109).
        Cloned object 93 to partition SAbck1 (new handle 110).
        Cloned object 49 to partition SAbck1 (new handle 111).
        Cloned object 100 to partition SAbck1 (new handle 112).
        Cloned object 25 to partition SAbck1 (new handle 113).
        Cloned object 47 to partition SAbck1 (new handle 114).
        Cloned object 62 to partition SAbck1 (new handle 115).
        Cloned object 51 to partition SAbck1 (new handle 118).
        Cloned object 113 to partition SAbck1 (new handle 119).
        Cloned object 106 to partition SAbck1 (new handle 120).
        Cloned object 58 to partition SAbck1 (new handle 121).
        Cloned object 102 to partition SAbck1 (new handle 124).
        Cloned object 70 to partition SAbck1 (new handle 125).
        Cloned object 78 to partition SAbck1 (new handle 128).
        Cloned object 88 to partition SAbck1 (new handle 129).
        Cloned object 40 to partition SAbck1 (new handle 130).

        Backup Complete.

        85 objects have been backed up to partition SAbck1
        on slot 3.

Command Result : No Error
```

The backup operation is complete. See below for an example of restoring from backup.

Restore to a SafeNet Luna PCIe HSM Slot

If your primary HSM partition (the partition onto which you will restore the backed-up objects) is in Activated state, then only the Backup HSM needs PED activity for authentication during restore. However, we add a couple of steps below to show that it is straightforward to use the single Remote PED with both HSMs, in the

case where your HSM partition is not in Activated state when you begin the restore operation.

1. For the example, start by clearing the target partition before restoring objects into it, so it is obvious that any objects after the restore operation are, in fact, restored, and not left-overs. This example is a replacement restore operation, and not an appending or cumulative restore operation.

```
lunacm:> partition clear

        You are about to delete all the user objects.
        Are you sure you wish to continue?

        Type 'proceed' to continue, or 'quit' to quit now -> proceed

        85 objects were deleted.

Command Result : No Error

lunacm:> exit
```

2. In our test setup, we have each of several SafeNet Luna PCIe HSM products. An easy way to see an updated summary of all HSMs and slot assignments is to exit LunaCM and restart the utility.

```
C:\Program Files\SafeNet\LunaClient>lunacm.exe

LunaCM V7.0.0 - Copyright (c) 2006-2016 Gemalto, Inc.


        Available HSM's:

        Slot Id ->              1
        HSM Label ->            SA82_P1
        HSM Serial Number ->    500409014
        HSM Model ->            LunaSA
        HSM Firmware Version -> 6.10.1
        HSM Configuration ->     SafeNet Luna PCIe HSM Slot (PED) Signing With Cloning Mode
        HSM Status ->           OK

        Slot Id ->              2
        HSM Label ->            G5PKI
        HSM Serial Number ->    701968008
        HSM Model ->            LunaSA
        HSM Firmware Version -> 6.10.1
        HSM Configuration ->     SafeNet Luna PCIe HSM Slot (PED) Signing With Cloning Mode
        HSM Status ->           OK

        Slot Id ->              3
        HSM Label ->            G5backup
        HSM Serial Number ->    700101
        HSM Model ->            G5Backup
        HSM Firmware Version -> 6.10.1
        HSM Configuration ->    Remote Backup HSM (PED) Backup Device
        HSM Status ->           OK

        Slot Id ->              4
        Tunnel Slot Id ->       6
        HSM Label ->            PCI422
        HSM Serial Number ->    500422
```

```
HSM Model ->            K6 Base
HSM Firmware Version -> 6.2.1
HSM Configuration ->    Luna PCI (PED) Signing With Cloning Mode
HSM Status ->           OK

Slot Id ->              5
Tunnel Slot Id ->       7
HSM Label ->            K6_328
HSM Serial Number ->    155328
HSM Model ->            K6 Base
HSM Firmware Version -> 6.10.1
HSM Configuration ->    Luna PCI (PED) Signing With Cloning Mode
HSM Status ->           OK

Slot Id ->              8
HSM Label ->            G5180
HSM Serial Number ->    700180
HSM Model ->            G5Base
HSM Firmware Version -> 6.10.1
HSM Configuration ->    SafeNet Luna USB HSM (PED) Signing With Cloning Mode
HSM Status ->           OK


Current Slot Id: 1
```

3. Verify which slot is listening for PED and whether it is expecting local or remote:

```
lunacm:>ped get

        HSM slot 1 listening to local PED (PED id=0).

Command Result : No Error
```

4. Connect to Remote PED:

```
lunacm:> ped connect ip 192.20.10.190

Command Result : No Error
```

5. Deactivate (just to demonstrate using PED with both HSMs):

```
lunacm:> partition deactivate

Command Result : No Error
```

6. Log into the partition. This would not be necessary if the partition was activated - we are demonstrating that if the partition was not in login state or activated state, it is straightforward to briefly switch the PED to the primary HSM partition before switching the PED back to the Backup HSM.

```
lunacm:> partition login

        Option -password was not supplied.  It is required.

        Enter the password: *******

        User is not activated, please attend to the PED.
```

```
Command Result : No Error

lunacm:> ped disconnect

        Are you sure you wish to disconnect the remote ped?

        Type 'proceed' to continue, or 'quit' to quit now -> proceed

Command Result : No Error
```

7. Now, (re)connect the Remote PED to the Backup HSM (which, in this example, is slot 3).

```
lunacm:> ped connect ip 192.20.10.190 slot 3

Command Result : No Error

lunacm:> ped get

        HSM slot 1 listening to local PED (PED id=0).

Command Result : No Error

lunacm:> ped get slot 3

        HSM slot 3 listening to remote PED (PED id=100).

Command Result : No Error
```

8. The currently selected slot is "slot 1" (see the LunaCM startup summary, above).
Now restore to the current slot from the slot that corresponds to the Backup HSM (slot 3).

```
lunacm:> partition backup restore -slot 3 -par SAbck1

        Logging in to partition SAbck1 on slot 3 as the user.

        Please attend to the PED.

        Verifying that all objects can be restored...

        85 objects will be restored.

        Restoring objects...
        Cloned object 19 from partition SAbck1 (new handle 20).
        Cloned object 20 from partition SAbck1 (new handle 21).
        Cloned object 23 from partition SAbck1 (new handle 22).
        Cloned object 25 from partition SAbck1 (new handle 25).
        Cloned object 24 from partition SAbck1 (new handle 26).
        Cloned object 26 from partition SAbck1 (new handle 28).
        Cloned object 28 from partition SAbck1 (new handle 29).
        Cloned object 27 from partition SAbck1 (new handle 30).
        Cloned object 29 from partition SAbck1 (new handle 33).
        Cloned object 30 from partition SAbck1 (new handle 34).
        Cloned object 31 from partition SAbck1 (new handle 40).
        Cloned object 35 from partition SAbck1 (new handle 44).
        Cloned object 36 from partition SAbck1 (new handle 45).
        Cloned object 39 from partition SAbck1 (new handle 48).
        Cloned object 40 from partition SAbck1 (new handle 49).
        Cloned object 44 from partition SAbck1 (new handle 53).
```

```
Cloned object 45 from partition SAbck1 (new handle 54).
Cloned object 46 from partition SAbck1 (new handle 55).
Cloned object 47 from partition SAbck1 (new handle 56).
Cloned object 48 from partition SAbck1 (new handle 57).
Cloned object 49 from partition SAbck1 (new handle 58).
Cloned object 50 from partition SAbck1 (new handle 59).
Cloned object 51 from partition SAbck1 (new handle 60).
Cloned object 52 from partition SAbck1 (new handle 61).
Cloned object 53 from partition SAbck1 (new handle 62).
Cloned object 56 from partition SAbck1 (new handle 65).
Cloned object 57 from partition SAbck1 (new handle 66).
Cloned object 58 from partition SAbck1 (new handle 67).
Cloned object 59 from partition SAbck1 (new handle 68).
Cloned object 60 from partition SAbck1 (new handle 69).
Cloned object 61 from partition SAbck1 (new handle 70).
Cloned object 62 from partition SAbck1 (new handle 71).
Cloned object 63 from partition SAbck1 (new handle 72).
Cloned object 64 from partition SAbck1 (new handle 73).
Cloned object 65 from partition SAbck1 (new handle 74).
Cloned object 66 from partition SAbck1 (new handle 75).
Cloned object 70 from partition SAbck1 (new handle 79).
Cloned object 71 from partition SAbck1 (new handle 80).
Cloned object 72 from partition SAbck1 (new handle 81).
Cloned object 73 from partition SAbck1 (new handle 82).
Cloned object 74 from partition SAbck1 (new handle 83).
Cloned object 75 from partition SAbck1 (new handle 84).
Cloned object 76 from partition SAbck1 (new handle 85).
Cloned object 77 from partition SAbck1 (new handle 86).
Cloned object 78 from partition SAbck1 (new handle 87).
Cloned object 79 from partition SAbck1 (new handle 88).
Cloned object 80 from partition SAbck1 (new handle 89).
Cloned object 81 from partition SAbck1 (new handle 90).
Cloned object 82 from partition SAbck1 (new handle 91).
Cloned object 83 from partition SAbck1 (new handle 92).
Cloned object 84 from partition SAbck1 (new handle 93).
Cloned object 86 from partition SAbck1 (new handle 94).
Cloned object 85 from partition SAbck1 (new handle 95).
Cloned object 87 from partition SAbck1 (new handle 96).
Cloned object 88 from partition SAbck1 (new handle 97).
Cloned object 89 from partition SAbck1 (new handle 98).
Cloned object 91 from partition SAbck1 (new handle 99).
Cloned object 90 from partition SAbck1 (new handle 100).
Cloned object 92 from partition SAbck1 (new handle 101).
Cloned object 96 from partition SAbck1 (new handle 105).
Cloned object 97 from partition SAbck1 (new handle 106).
Cloned object 98 from partition SAbck1 (new handle 107).
Cloned object 99 from partition SAbck1 (new handle 108).
Cloned object 103 from partition SAbck1 (new handle 112).
Cloned object 104 from partition SAbck1 (new handle 113).
Cloned object 105 from partition SAbck1 (new handle 114).
Cloned object 106 from partition SAbck1 (new handle 115).
Cloned object 107 from partition SAbck1 (new handle 116).
Cloned object 108 from partition SAbck1 (new handle 117).
Cloned object 110 from partition SAbck1 (new handle 118).
Cloned object 109 from partition SAbck1 (new handle 119).
Cloned object 111 from partition SAbck1 (new handle 120).
Cloned object 112 from partition SAbck1 (new handle 121).
Cloned object 113 from partition SAbck1 (new handle 122).
```

```
        Cloned object 114 from partition SAbck1 (new handle 123).
        Cloned object 115 from partition SAbck1 (new handle 124).
        Cloned object 118 from partition SAbck1 (new handle 127).
        Cloned object 119 from partition SAbck1 (new handle 128).
        Cloned object 120 from partition SAbck1 (new handle 129).
        Cloned object 121 from partition SAbck1 (new handle 130).
        Cloned object 124 from partition SAbck1 (new handle 133).
        Cloned object 125 from partition SAbck1 (new handle 134).
        Cloned object 128 from partition SAbck1 (new handle 137).
        Cloned object 129 from partition SAbck1 (new handle 138).
        Cloned object 130 from partition SAbck1 (new handle 139).

        Restore Complete.

        85 objects have been restored from partition SAbck1 on slot 3.

Command Result : No Error
```

9. Verify that the restored slot now looks like it did just before the backup was originally performed.

```
lunacm:> partition backup list -slot 3

        HSM Storage Information for slot 3:

           Total HSM Storage Space:      16252928
           Used HSM Storage Space:       43616
           Free HSM Storage Space:       16209312
           Number Of Allowed Partitions: 20
           Number Of Allowed Partitions: 1

        Partition list for slot 3

           Number of partition: 1

           Name:                      SAbck1
           Total Storage Size:        41460
           Used Storage Size:         41460
           Free Storage Size:         0
           Number Of Objects:         85

Command Result : No Error
```

Restore from backup, using RBS, is complete.

To restore onto a different remote SafeNet Luna PCIe HSM, the same arrangement is required, but the remote HSM must already have a suitable partition (if the restore-target HSM is a SafeNet Luna PCIe HSM, the target partition can have any name - it does not need to match the name of the source partition on the backup device), and your workstation must be registered as a client to that partition.

To restate: the backup operation can go from a source partition (on a SafeNet Luna PCIe HSM) to an existing partition on the SafeNet Luna Backup HSM, or if one does not exist, a new partition can be created during the backup. But the restore operation cannot create a target partition on a SafeNet Luna PCIe HSM; it must already exist and have a registered NTLS link.

Restore your HSM Partition Locally

The options, in restoring to a partition are:

- To restore from a backup partition on a SafeNet Luna Backup HSM (modern backup and restore).
- To restore from a legacy backup token in a SafeNet DOCK slot (legacy restore, one way only, using legacy domain).

To restore one HSM Partition, you must have:

- The SafeNet Luna Backup HSM (also called Token) containing the objects to be restored to that partition
- The authentication for the Backup HSM and for the HSM Partition

The Backup Token and the HSM with the target partition must share the same cloning domain.

If you have Private Key Cloning switched off for the current partition, then the Backup operation proceeds, but skips over any private keys, and clones only the permitted objects onto the Backup token. Similarly, if you restore from a token that includes private keys, but the target partition has Private Key Cloning disallowed, then all other objects are recovered to the partition, but the private keys are skipped during the operation.

1. Insert a SafeNet Backup token into the token-reader slot on the SafeNet appliance front panel.
2. Choose an HSM Partition, and type:

```
partition restore -partition HSMPartitionname -password ClientPassword -replace
```

> **NOTE** In the command above, you could have used **-add** instead of **-replace**.

Example – partition restore Command

```
lunash:> partition restore -partition myRoom -password 9YWt6L56FXqGC6sL -replace
```

In that example, either the password came from the Luna PED of a SafeNet Luna PCIe HSM with Trusted Path Authentication, or it was a Password Authentication Partition Password created by someone very enthusiastic about passwords.

On restore, you may **add** to existing HSM Partition contents or **replace** them. Adding may result in unwanted behaviors, such as having two keys with the same label, if one existed in the HSM Partition and one on the backup token.

Restore your HSM Partition from Token

A SafeNet Luna PCIe HSM 5.x HSM can have up to 20 partitions, with space for objects per HSM defaulting to 2MB, upgradable to 15.5MB. Each partition on the HSM has a share of that space and can have its own cloning domain as represented by a domain (red) PED key.

The normal backup-and-restore option for SafeNet Luna PCIe HSM 5 partitions uses the external, locally connected or remotely linked (network) SafeNet Luna Backup HSM as the backup repository. The SafeNet Luna Backup HSM supports the same partition structure, storage size, and capacity as the SafeNet Luna PCIe HSM 5's onboard HSM.

In order to provide a migration path from earlier SafeNet Luna PCIe HSM and removable-token format HSMs, it is possible to externally connect a SafeNet DOCK 2 card reader for SafeNet PCM, SafeNet CA4, or SafeNet Luna HSM Backup Token, and to restore/migrate legacy token and partition contents to the current-generation SafeNet Luna PCIe HSM.

Keys (objects) from multiple SafeNet CA4 tokens, SafeNet PCM tokens (Key Export Signing, RA), or with differing cloning domains can be consolidated onto one SafeNet Luna PCIe HSM 5.x HSM, where objects from every token HSM are restored onto a partition corresponding to each token (segregated by legacy cloning domain). So, for example, ten legacy tokens (each with 100 objects) go to ten SafeNet Luna PCIe HSM partitions to accommodate however-many objects existed on all those tokens. The SafeNet Luna PCIe HSM in this example receives 1000 objects, allocated as 100 per partition, with each token migrating to its own SafeNet Luna PCIe HSM partition.

Alternatively, multiple SafeNet CA4 tokens, SafeNet PCM tokens, or SafeNet Luna PCIe HSM Backup Tokens can be restored to the same partition if those SafeNet CA4 (or Backup) tokens share the same domain PED key. So, for example, objects from ten tokens (each with 100 objects) go all on one partition which, at the end of the operation, contains 1000 objects.

Requirements

To restore an HSM partition from a removable token (firmware 4.x), to a SafeNet Luna PCIe HSM 5.x HSM, you must have:

- The SafeNet Backup Token containing the objects to be restored to that HSM
- The authentication (the authentication type must match - if your source tokens are password-authenticated, their contents can be restored/migrated onto a password-authenticated HSM partition only; if your source tokens are PED-authenticated, their contents can be restored/migrated onto a PED-authenticated HSM partition only) for the Backup Token or PCM token, and for the HSM Partition
- SafeNet DOCK 2 card reader

The types of objects that can be migrated also depend on the configurations and policies of the source and destination HSMs. For example, the RA (registration authority) configuration permits cloning of secret keys, but not of private keys, and that intentional, security policy-based limitation applies to the migration/restore-from-legacy operation as well.

In the following examples, the target, or destination partition is called mylunapar2.

For SafeNet Luna PCIe HSM with Password authentication:

1. Create a partition on the SafeNet Luna PCIe HSM 5 HSM :

   ```
   lunash:>partition create -partition mylunapar2 -password <password> -domain <domain> -force
   ```

2. With the SafeNet DOCK 2 reader powered on and connected (USB) to the SafeNet Luna PCIe HSM 5.x, insert a SafeNet Luna PCIe HSM Backup token (or other legacy removable token-format HSM) into the token-reader slot of the SafeNet DOCK.
3. Type the command:

   ```
   lunash:>partition restore -password mylunapar2 -password <password> [-tokenPar <name>] [-
   tokenPw <tokenpassword>] -add
   ```

For SafeNet Luna PCIe HSM with PED authentication:

1. Create a partition on the SafeNet Luna PCIe HSM 5 HSM:

   ```
   lunash:>partition create -partition mylunapar2 -force
   ```

   Both user (black) and domain (red) PED keys are created for SafeNet Luna PCIe HSM 5 partition mylunapar2.

2. With the SafeNet DOCK 2 reader powered on and connected (USB) to your client computer, insert the desired SafeNet Luna PCIe HSM Backup token or SafeNet CA4 token into the token-reader slot of the SafeNet DOCK 2.

3. Leave the SafeNet DOCK 2 powered on and the token in its slot, and transfer its USB cable connection from the client computer to the USB socket on the SafeNet Luna PCIe HSM 5.x. The SafeNet Luna PCIe HSM immediately sees the new token slot, and you can now run LunaSH commands from the SafeNet Luna PCIe HSM against the token.

4. Import the legacy domain:

   ```
   lunash:>partition setLegacyDomain -partition mylunapar2 [-password <password>] [-domain
   <domain>]
   ```

   and respond to the PED prompts including presenting the legacy red key.

   SafeNet Luna PCIe HSM 5, SafeNet Luna USB HSM, and the SafeNet Remote Backup Device use a newer domain scheme, which is not compatible with legacy HSM domains. The **partition setLegacyDomain** command prepares a legacy domain in a way that allows it to be recognized and used by a current-model HSM, in special circumstances; the HSM retains its modern domain, but the legacy domain becomes associated with the partition's "real" domain. The association is permanent for the life of that partition. Intentional,designed-in, data security provisions prevent setting/associating a legacy domain from one SafeNet token to a single SafeNet Luna PCIe HSM 5.x partition, then associating another (different) legacy domain to that same partition and adding the second token's objects to the partition while the first token's objects are stored there. Just as you cannot clone/copy objects from one token to another token with a different domain, you cannot get around that security provision by migrating unmatched domain objects to a single SafeNet Luna PCIe HSM partition.
   As long as token HSMs share a common (legacy) domain, you migrate the contents of multiple tokens to a single partition - the legacy domain is set just once for all such tokens.

5. Type the command:

   ```
   lunash:>partition restore -partition mylunapar2 -replace -force
   ```

   and respond to the PED prompts. The **-replace** option overwrites the partition content with objects from the SafeNet CA4 or PCM or Backup token. Use the **-add** option if you want to append the SafeNet token objects to the partition.

6. Repeat all the above steps to restore objects from other SafeNet tokens onto separate SafeNet Luna PCIe HSM partitions.
   Repeat only step 6 with the **-add** option, instead, to restore objects from other SafeNet tokens onto the same, single SafeNet Luna PCIe HSM 5 partition - this works only if the originating SafeNet tokens all share the same legacy domain. Once a legacy domain is associated with a SafeNet Luna PCIe HSM 5.x partition, that association remains in force for the life of that partition; the HSM does not allow another association (of

legacy domain) to be made onto a partition that already has an existing association. The only way to end the association is to destroy the partition (wiping all contents) and create it again.

If you have a PED authenticated token HSM, but did not have MofN authentication applied, then the steps are the same as above except you do not issue the LunaCM **partition mofnactivate** command.

Backing up the HSM contents to a token-style HSM is not a supported operation for SafeNet Luna PCIe HSM 5.x.

Restore from a legacy backup token is effectively a data migration - one-way only.

Troubleshooting and Frequently Asked Questions

This section contains answers to frequently asked questions or known errors.

> "Troubleshooting "token not in factory reset state" Error" on the next page

> "Backup HSM Battery Questions" on page 79

Why is Backup optional?

In general, a SafeNet Luna PCIe HSM or HSM Partition is capable of being backed up to a SafeNet Luna Backup HSM via cloning (the **partition backup** command uses cloning functionality to securely copy objects from the source HSM to a target Backup HSM), depending on the configuration variant you have purchased. The backup capability is considered a good and desirable and necessary thing for keys that carry a high cost to replace, such as Certificate Authority root keys and root certificates.

However, Backup HSMs are optional equipment for SafeNet Luna PCIe HSMs. There are at least two reasons for this:

1. Some Customers don't care. They may be using (for example) SSL within a controlled boundary like a corporation, where it is not a problem to simply tell all employees to be prepared to trust a new certificate, in the event that the previous one is lost or compromised. In fact it might be company policy to periodically jettison old certificates and distribute fresh ones.
   Other customers might be using software that manages lost profiles, making it straightforward to resume work with a new key or certificate. The certificate authority that issued the certificates would need backup, but the individual customers of that certificate authority would not.

   In summary, it might not be worthwhile to backup keys that are low-cost (from an implementation point of view) to replace. Keys that carry a high cost to replace should be backed up.
2. Some countries do not permit copying of private keys. If you are subject to such laws, and wish to store encrypted material for later retrieval (perhaps archives of highly sensitive files), then you would use symmetric keys, rather than a private/public key-pair, for safe and legal backup.

How long does data last?

SafeNet Luna PCIe HSMs have onboard volatile memory meant for temporary data (disappears when power is removed), and onboard flash memory, used to store permanent material, like PKI Root keys and other critical key material, and like the firmware that makes the device work.

No electronic storage is forever. If your SafeNet Luna PCIe HSM is operated within an ambient temperature range of 0 degrees Celsius to +40 degrees Celsius, or stored between -20 degrees Celsius and +65 degrees Celsius, then (according to industry-standard testing and estimation methods) your data should be retrievable for twenty years from the time that the token was shipped from the factory. This is a conservative estimate, based on worst-case characteristics of the system components.

What does this mean to me?
Advances in technology will probably ensure that you never need to test the expected expiration of data on your SafeNet Luna PCIe HSMs.

Troubleshooting "token not in factory reset state" Error

If you insert a backup token that has previously been used on a Password Authenticated SafeNet Luna PCIe HSM into a PED Authenticated SafeNet Luna PCIe HSM, and attempt to initialize it, the system presents an error like:

```
[mylunasa] lunash:>token backup init -label mylunatoken  -serial 1234567 -force


Warning:  This token is not in the factory reset (zeroized) state.

          You must present the current Token Admin login credentials

          to clear the backup token's contents.

Luna PED operation required to initialize backup token - use
         Security Officer (blue) PED key.

Error:  'token init' failed. (300130 : LUNA_RET_INVALID_ENTRY_TYPE)


Command Result : 65535 (Luna Shell execution)

[mylunasa] lunash:>
```

This is a security feature, intended to prevent backup of PED-secured HSM objects onto a less secure Password Authenticated token.
To work around this problem, issue **token factoryReset**, and then initialize the token:

```
[mylunasa] lunash:>token backup factoryReset   -serial 1234567
CAUTION:  Are you sure you wish to reset this backup token to

          factory default settings? All data will be erased.

         Type 'proceed' to return the token to factory default, or

         'quit' to quit now.

         > proceed

token factoryReset' successful.
Command Result : 0 (Success)

[mylunasa] lunash:>token backup init -label mylunatoken   -serial 1234567  -force
        Luna PED operation required to initialize backup token - use
```

```
        Security Officer (blue) PED key.
        Luna PED operation required to login to backup token - use
        Security Officer (blue) PED key.
        Luna PED operation required to generate cloning domain on
         backup token - use Domain (red) PED key.

'token init' successful.

Command Result : 0 (Success)
[mylunasa] lunash:>
```

Comparison Summary

See "Comparison of Destruction/Denial Actions" on page 121 to view a table that compares and contrasts various "deny access" events or actions that are sometimes confused.

Backup HSM Battery Questions

The SafeNet Luna Backup HSM (for backing up and restoring HSM and partition contents) can be stored, with valuable contents, when not in use.

The battery that powers the NVRAM and RTC must be installed for use, but some questions commonly arise if the device is to be stored for long periods. As an administrator of HSMs, I need clear instructions on what to do/how to manage the battery in the SafeNet Luna USB HSM and SafeNet Luna Backup HSM so that I don't get into a situation where I can't retrieve my backups or use my HSM.

Should I take the battery out when storing the HSM in a safe?

It is generally good practice to remove batteries when storing electronic devices, to preclude accidental damage from battery leakage. We use high-quality, industrial-grade batteries, that are unlikely to fail in a damaging fashion, but prudence suggests removing them, regardless. Also, if the unit is not in use, there is no need to maintain power to the RTC and NVRAM, so an externally stored battery will last longer (see specifications, below).

If the battery is out, what happens?

If main power is not connected, and the battery dies, or is removed, then NVRAM and the system's Real Time Clock lose power. The working copy of the MTK is lost.

If the battery dies during operation, will I lose my key material? Will corruption occur?

The only key material that is lost is temporary session objects (including working copies of stored keys) that are in use at the time. If the "originals" of those same objects are stored as HSM/partition objects, then they reside in non-volatile memory, and those are preserved.

There is no corruption of stored objects.

Where can I get a spare/replacement battery?

From any supplier that can match the specifications.

Technical Specs

3.6 V Primary lithium-thionyl chloride (Li-SOCl2)

Fast voltage recovery after long term storage and/or usage

Low self discharge rate

10 years shelf life

Operating temperature range -55 ºC to +85 ºC

U.L. Component Recognition, MH 12193

Storage Conditions

Cells should be stored in a clean & dry area (less than 30 % Relative Humidity)

Temperature should not exceed +30 ºC

How do I know if the battery is dead or about to die? Can I check the status of the battery?

There is not a low battery indicator or other provision for checking status.

The battery discharge curve is such that the voltage remains constant until the very end of the battery life, at which point the discharge is extremely steep.

What must I do to recover function, and access to my key material, after battery removal/discharge?

Simply insert the battery, connect the HSM, power it up, and resume using it.

The MTK that was deleted by the tamper event (battery removal/discharge) is reconstituted from stored portions as soon as you log in. All your stored material is available for use.

CHAPTER 3: Capabilities and Policies

The SafeNet Luna PCIe HSM's configuration is based on HSM capabilities, displayed by issuing the command **hsm showpolicies** on the Admin partition. They are set at manufacture according to the model you selected at time of purchase. Capabilities can only be modified by purchase and application of capability updates.

A subset of HSM capabilities have corresponding HSM policies that allow you to customize the HSM configuration. Policies can be modified based on your specific needs. For example, you can restrict the HSM to use only FIPS-approved algorithms (FIPS mode) by setting HSM policy **12** to 1 (on).

Partitions inherit the capabilities and policy settings of the HSM. Partitions also have policies that can be set to customize the partition functions. Partition policies can never be modified to be less secure than the corresponding HSM capability/policy. For example, if HSM policy **7** is set to disallow cloning, partition policies **0** and **4**, which allow cloning of private or secret keys, cannot be set to 1 (on).

The following sections describe individual HSM/partition capabilities and policies:

> "HSM Capabilities and Policies" below

> "Partition Capabilities and Policies" on page 87

The HSM or Partition SO can create and apply Policy Templates to initialize multiple HSMs/partitions with the same preferred policy settings. See the following section for instructions on using Policy Templates:

> "Policy Templates" on page 93

HSM Capabilities and Policies

The SafeNet Luna PCIe HSM's configuration is based on HSM capabilities. They are set at manufacture according to the model you selected at time of purchase. Capabilities can only be modified by purchase and application of capability updates.

A subset of HSM capabilities have corresponding HSM policies that allow you to customize the HSM configuration. Policies can be modified based on your specific needs. They can never be modified to be less secure than the corresponding capability.

To view the HSM capability and policy settings, use the LunaSH command **hsm showpolicies**. Include the **-exporttemplate** option to create a template based on the current HSM policy settings. See "Policy Templates" on page 93.

To modify HSM policies, log in as HSM SO and use the LunaSH command **hsm changepolicy -policy** <policy#> **-value** <0/1>. See "hsm changepolicy" on page 1 in the *LunaSH Command Reference Guide* for command syntax.

To zeroize the HSM and reset the policies to their default values, use **hsm factoryreset**. See "hsm factoryreset" on page 1 in the *LunaSH Command Reference Guide* for command syntax.

To zeroize the HSM and keep the current policy settings, use **hsm zeroize**. See "hsm zeroize" on page 1 in the *LunaSH Command Reference Guide* for command syntax.

To view the HSM capability and policy settings, issue the LunaCM command **hsm showpolicies** on the Admin partition. Only policies that the HSM SO can change (the corresponding capability is not set to **0**) are included in the output. Include the **-exporttemplate** option to create a template based on the current HSM policy settings. See "Policy Templates" on page 93.

To modify HSM policies, log in as HSM SO and use the LunaCM command **hsm changehsmpolicy-policy** <policy#> **-value** <0/1>. See "hsm changehsmpolicy" on page 1 in the *LunaCM Command Reference Guide* for command syntax.

To zeroize the HSM and reset the policies to their default values, use **hsm factoryreset**. See "hsm factoryreset" on page 1 in the *LunaCM Command Reference Guide* for command syntax.

To zeroize the HSM and keep the current policy settings, use **hsm zeroize**. See "hsm zeroize" on page 1 in the *LunaCM Command Reference Guide* for command syntax.

Destructiveness

In some cases, changing an HSM policy zeroizes all application partitions or the entire HSM as a security measure. These policies are listed as **destructive** in the table below.

HSM Capability and Policy Descriptions

The table below summarizes the relationships and provides a brief description of the purpose and operation of each capability and policy.

#	HSM Capability	HSM Policy	Description
0	Enable PIN-based authentication		If allowed, the HSM authenticates all users with keyboard-entered passwords.
1	Enable PED-based authentication		If allowed, the HSM authenticates users with secrets stored on physical PED keys, read by a SafeNet Luna PED. The Crypto Officer and Crypto User roles may also be configured with a secondary, keyboard-entered challenge secret.
2	Performance level		Numerical value indicates the performance level of this HSM, determined by the model you selected at time of purchase: > **4: Standard** performance > **8: Enterprise** performance > **15: Maximum** performance
4	Enable domestic mechanisms & key sizes		Always allowed. All SafeNet Luna HSMs are capable of full-strength cryptography with no US export restrictions.
6	Enable masking		Always disallowed. SIM has been deprecated on all current SafeNet Luna PCIe HSMs.

#	HSM Capability	HSM Policy	Description
7	Enable cloning	Allow cloning	If allowed, the HSM is capable of cloning cryptographic objects from one partition to another. This policy must be enabled to backup partitions over a network or create HA groups. Partition Security Officers may then enable/disable cloning on individual partitions. **Destructive: OFF-to-ON**
9	Enable full (non-backup) functionality		If allowed, the HSM is capable of full cryptographic functions. This capability is only disallowed on SafeNet Luna Backup HSMs.
12	Enable non-FIPS algorithms	Allow non-FIPS algorithms	If allowed, the HSM can use all available cryptographic algorithms. If disallowed, only algorithms sanctioned by the FIPS 140-2 standard are permitted. The following is displayed in the output from **hsm showinfo** in LunaCM: `The HSM is in FIPS 140-2 approved operation mode.` **Destructive: OFF-to-ON**
15	Enable SO reset of partition PIN	SO can reset partition PIN	If allowed, a Partition SO can reset the password or PED secret of a Crypto Officer who has been locked out after too many bad login attempts. If disallowed, the lockout is permanent and the partition contents are no longer accessible. The partition must be re-initialized, and key material restored from a backup device. See "Failed Login Attempts" on page 329 for more information. **Destructive: OFF-to-ON, ON-to-OFF**
16	Enable network replication	Allow network replication	If allowed, cryptographic object cloning is permitted over a network. This is required for HA groups, and for partition backup to a remote or client-connected SafeNet Luna Backup HSM. If disallowed, cloning over a network is not permitted. Partition backup is possible to a locally-connected SafeNet Luna Backup HSM only. Setting this policy to **0** means that only the HSM SO can backup partitions.
17	Enable Korean Algorithms	Allow Korean algorithms	If allowed, the SafeNet Luna PCIe HSM can use the Korean algorithm set. This capability may be purchased as an upgrade. See "Upgrading HSM Capabilities" on page 320.
18	FIPS evaluated		Always disallowed - deprecated policy. All SafeNet Luna PCIe HSMs are capable of operating in FIPS Mode.
19	Manufacturing Token		N/A (SafeNet internal use only)

#	HSM Capability	HSM Policy	Description
21	Enable forcing user PIN change	Force user PIN change after set/reset	If allowed, when a Partition SO initializes the Crypto Officer role (or resets the password/PED secret), the CO must change the credential with **role changepw** before any other actions are permitted. The same is true when the CO initializes/resets the Crypto User role. This policy is intended to enforce the separation of roles on the partition. If disallowed, the CO/CU may continue to use the credential assigned by the Partition SO.
22	Enable offboard storage	Allow off-board storage	On previous HSMs, this policy allowed or disallowed the use of the portable SIM key. SIM is not supported on this version of SafeNet Luna HSM. **Destructive: OFF-to-ON**
23	Enable partition groups		Always disallowed - deprecated policy.
25	Enable Remote PED usage	Allow Remote PED usage	Always enabled on PED-authenticated SafeNet Luna PCIe HSMs. All PED-authenticated HSMs are capable of connecting to a local PED or a remotely-located PED server. The HSM SO may turn this feature on or off.
27	HSM non-volatile storage space		Displays the non-volatile maximum storage space (in bytes) on the HSM. This is determined by the model of SafeNet Luna PCIe HSM you selected at time of purchase.
30	Enable Unmasking	Allow unmasking	If allowed, cryptographic material can be migrated from legacy SafeNet appliances that used SIM.
33	Maximum number of partitions	Current maximum number of partitions	Displays the maximum number of application partitions that can be created on the HSM. This number is determined by the model of SafeNet Luna PCIe HSM you selected at time of purchase. On some models, the number of allowable partitions can be upgraded with a separate purchase.
35	Enable Single Domain		Not applicable to SafeNet Luna PCIe HSMs.
36	Enable Unified PED Key		Not applicable to SafeNet Luna PCIe HSMs.

#	HSM Capability	HSM Policy	Description
37	Enable MofN	Allow MofN	If allowed on PED-authenticated SafeNet Luna PCIe HSMs, this policy enables you to require a quorum for role access, by splitting a PED secret among multiple PED keys (see "M of N Split Secrets (Quorum)" on page 192). If disallowed, users will no longer be asked to split a PED secret (M and N automatically set to 1). Always disallowed on password-authenticated HSMs.
38	Enable small form factor backup/restore		Not available in this release.
39	Enable Secure Trusted Channel	Allow Secure Trusted Channel	If allowed, this policy enables the use of Secure Trusted Channel for partition-client connections (see "Secure Trusted Channel (STC)" on page 1). If disallowed, all partition-client connections must use NTLS. Secure Trusted Channel is a Network HSM feature, and has no function on SafeNet Luna PCIe HSM. Thales Group does not recommend turning this policy on at any time.
40	Enable decommission on tamper	Decommission on tamper	If allowed, the HSM will be decommissioned if a tamper event occurs. Decommissioning deletes all partitions and their contents, the audit role, and the audit configuration. The HSM policy settings are retained. See "Tamper Events" on page 281 for more information. **Destructive: ON-to-OFF**
42	Enable partition re-initialize		Not available in this release.
43	Enable low level math acceleration	Allow low-level math acceleration	This is enabled by default, and must be enabled to provide maximum performance. Do not disable unless instructed to do so by Thales Group Technical Support.
46	Allow Disabling Decommission	Disable Decommission	If enabled, the decommission jumper header is disabled, preventing decommissioning of the HSM. **CAUTION:** Changing this policy will destroy partitions on the HSM, and they must be recreated. If HSM policy 40: Decommission on Tamper is enabled, you cannot enable this policy (fails with error: CKR_CONFIG_FAILS_DEPENDENCIES). However, attempting to enable it will still destroy HSM partitions. **Destructive: OFF-to-ON, ON-to-OFF**

#	HSM Capability	HSM Policy	Description
47	Enable Tunnel Slot		Not available in this release.
48	Enable Controlled Tamper Recovery	Do Controlled Tamper Recovery	If allowed, the HSM SO must explicitly clear the tamper before the HSM can resume normal operations. This is the default behavior. If disallowed, the HSM must be restarted before it can resume normal operations. See "Tamper Events" on page 281 for more information.
49	Enable Partition Utilization Metrics	Allow Partition Utilization Metrics	If allowed the HSM SO and Administrator can view (or export to a named file) counters that record how many times specific cryptographic operations have been performed in application partitions since the last counter-reset event. This provides a picture of operational utilization that can be used to guide the [re-]allocation and balancing of partitions and applications, for better service to all users of your partitions.

Partition Capabilities and Policies

Partitions inherit the capabilities and policy settings of the HSM. Partitions also have policies that can be set to customize the partition functions. Partition policies can never be modified to be less secure than the corresponding HSM capability/policy. For example, if the HSM's cloning policy is disallowed (see HSM policy **7**), partition policies **0** and **4**, which allow cloning of private or secret keys, cannot be set.

> **NOTE** If you are running more than one LunaCM session against the same partition, and change a partition policy in one LunaCM session, the policy change will be reflected in that session only. You must exit and restart the other LunaCM sessions to display the changed policy settings.

To view the partition capabilities and policy settings, use the LunaCM command **partition showpolicies**. Only policies that the Partition SO can change (the corresponding capability is not set to **0**) are included in the output. Include the **-exporttemplate** option to create a template based on the current partition policy settings. See "Policy Templates" on page 93.

To modify partition policies, login as Partition SO and use the LunaCM command **partition changepolicy -policy** <policy#> **-value** <0/1/value>. See "partition changepolicy" on page 1 in the *LunaCM Command Reference Guide* for command syntax.

Destructiveness

In some cases, changing a partition policy forces deletion of all cryptographic objects on the partition as a security measure. These policies are listed as **destructive** in the table below. Destructive policies are typically those that change the security level of the objects stored in the partition.

Use the LunaCM command **partition showpolicies -verbose** to check whether the policy you want to enable/disable is destructive.

Partition Capabilities and Policies List

The table below summarizes the relationships and provides a brief description of the purpose and operation of each capability and policy.

#	Partition Capability	Partition Policy	Description
0	Enable private key cloning	Allow private key cloning	If enabled, the partition is capable of cloning private keys to another partition. This policy must be enabled to backup partitions or create HA groups. Public keys/objects can always be cloned. Partition policies **0** and **1** may not be set to **1** (ON) at the same time. Default: **ON** **Destructive: OFF-to-ON**
1	Enable private key wrapping	Allow private key wrapping	If enabled, private keys may be wrapped and saved to an encrypted file off the partition. Public keys/objects can always be wrapped and exported. Partition policies **0** and **1** may not be set to **1** (ON) at the same time. Default: **OFF** **Destructive: OFF-to-ON**
2	Enable private key unwrapping	Allow private key unwrapping	If enabled, private keys may be unwrapped onto the partition. The Partition SO can turn this feature on or off. If disabled, private key unwrapping is not available, and the Partition SO cannot change this. Default: **ON**
3	Enable private key masking	Allow private key masking	Always disabled. SIM has been deprecated on all current SafeNet Luna PCIe HSMs. The Partition SO cannot change this policy. Default: always **OFF**
4	Enable secret key cloning	Allow secret key cloning	If enabled, secret keys on the partition can be backed up. The Partition SO can turn this feature on or off. If disabled, secret keys cannot be backed up, and the Partition SO cannot change this. Partition backup or partition network replication is allowed for the SafeNet high availability feature. Default: **ON** **Destructive: OFF-to-ON**
5	Enable secret key wrapping	Allow secret key wrapping	If enabled, secret keys can be wrapped off the partition. The Partition SO can turn this feature on or off. The Partition SO can turn this policy off to disallow secret key wrapping If disabled, the partition does not support secret key wrapping, and the Partition SO cannot change this. Default: **ON** **Destructive: OFF-to-ON**

#	Partition Capability	Partition Policy	Description
6	Enable secret key unwrapping	Allow secret key unwrapping	If enabled, secret keys can be unwrapped onto the partition. The Partition SO can turn this feature on or off. If disabled, the partition does not support secret key unwrapping, and the Partition SO cannot change this. Default: **ON**
7	Enable secret key masking	Allow secret key masking	Always disabled. SIM has been deprecated on all current SafeNet Luna PCIe HSMs. The Partition SO cannot change this policy. Default: always **OFF**
10	Enable multipurpose keys	Allow multipurpose keys	If enabled, keys that are created or unwrapped on the partition may have more than one of the following attributes set to **1**, and therefore can be used for multiple operations: > Encrypt/Decrypt > Sign/Verify > Wrap/Unwrap > Derive If disabled, keys on the partition may have only one of these attributes set to **1**. Thales Group recommends that you create keys with only the attributes required for their intended purpose. Disabling this policy enforces this rule on the partition. This policy does not affect Diffie-Hellman keys, which are always created with only Derive set to **1**. Default: **ON** **Destructive: OFF-to-ON**
11	Enable changing key attributes	Allow changing key attributes	If enabled, non-sensitive attributes of the keys on the partition are modifiable (the user can change the functions that the key can use). If disabled, keys created on the partition cannot be modified. This policy affects the following "key function attributes": CKA_ENCRYPT CKA_DECRYPT CKA_WRAP CKA_UNWRAP CKA_SIGN CKA_SIGN_RECOVER CKA_VERIFY CKA_VERIFY_RECOVER CKA_DERIVE CKA_EXTRACTABLE Default: **ON** **Destructive: OFF-to-ON**

#	Partition Capability	Partition Policy	Description
15	Allow failed challenge responses	Ignore failed challenge responses	This policy applies to PED-authenticated SafeNet Luna HSMs only. The Partition SO can turn the feature on or off. If enabled, failed challenge secret login attempts on an activated partition are not counted towards a partition lockout. Only failed PED key authentication attempts will increment the counter. If disabled, failed login attempts using either a PED key or a challenge secret will count towards a partition lockout. See "Activation and Auto-Activation on PED-Authenticated Partitions" on page 178 and "Failed Login Attempts" on page 329 for more information. Default: **ON** **Destructive: OFF-to-ON**
16	Enable operation without RSA blinding	Operate without RSA blinding	If enabled, the partition may run in a mode that does not use RSA blinding (a technique that introduces random elements into the signature process to prevent timing attacks on the RSA private key. Use of this technique may be required by certain security policies, but it does reduce performance). The Partition SO can turn this feature on or off. If disabled, the partition will always run in RSA blinding mode; performance will be affected. If the policy is set to **1** (ON), RSA blinding is not used. Default: **ON** **Destructive: OFF-to-ON**
17	Enable signing with non-local keys	Allow signing with non-local keys	If a key was generated on an HSM, CKA_LOCAL is set to 1. With this policy turned off, only keys with CKA_LOCAL=1 can be used to sign data on the HSM. Keys that are imported (unwrapped) to the HSM have CKA_LOCAL explicitly set to 0, so they may not be used for signing. Cloning and SIM maintain the value of CKA_LOCAL. With this policy turned on, keys that did not originate on the HSM (CKA_LOCAL=0) may be used for signing, and their trust history is not assured. Default: **ON**
18	Enable raw RSA operations	Allow raw RSA operations	If enabled, the partition may allow raw RSA operations (mechanism CKM_RSA_X_509). This allows weak signatures and weak encryption. The Partition SO can turn this feature on or off. If disabled, the partition will not support raw RSA operations. Default: **ON** **Destructive: OFF-to-ON**

#	Partition Capability	Partition Policy	Description
20	Max failed user logins allowed	Max failed user logins allowed	Displays the maximum number of failed partition login attempts before the partition is locked out (see "Failed Login Attempts" on page 329). The Partition SO can change the number of failed logins to a value lower than the maximum if desired. Default: **10**
21	Enable high availability recovery	Allow high availability recovery	If enabled, partitions in the same HA group may be used to restore the login state of this partition after power outage or other deactivation. RecoveryLogin must be configured in advance (see "role recoveryinit" on page 1 and "role recoverylogin" on page 1 in the *LunaCM Command Reference Guide* for details. The Partition SO can turn this feature on or off. Default: **ON**
22	Enable activation	Allow activation	Applies only to PED-authenticated HSMs. If enabled, the black and/or gray PED key secrets may be cached, so that the CO or CU only needs the challenge secret to login. The Partition SO can turn this feature on or off. If disabled (or the policy is turned off), PED keys must be presented at each login, whether the call is local or from a client application. This policy setting is overridden and activation is disabled if a tamper event occurs, or if an uncleared tamper event is detected on reboot. See "Tamper Events" on page 281, and "Activation and Auto-Activation on PED-Authenticated Partitions" on page 178 for more information. Default: **OFF**
23	Enable auto-activation	Allow auto-activation	See Capability 22 above for a description of activation. If enabled, the black or gray PED key secrets may be encrypted and semi-permanently cached to hard disk, so that the partition's activation status can be maintained after a power loss of up to two hours. The Partition SO can turn this feature on or off. If disabled, this partition does not support auto-activation. This policy setting is overidden and auto-activation is disabled if a tamper event occurs, or if an uncleared tamper event is detected on reboot. See "Tamper Events" on page 281, and "Activation and Auto-Activation on PED-Authenticated Partitions" on page 178 for more information. Default: **OFF**

#	Partition Capability	Partition Policy	Description
25	Minimum PIN length (inverted: 255 - min)	Minimum PIN length (inverted: 255 - min)	The absolute minimum length for a partition login PIN is 8 characters. This is displayed as a value subtracted from 256. The policy value is determined as follows: Subtract the desired minimum PIN length from 256 (the absolute maximum length), and set policy 25 to that value. **256 - (min PIN) = (policy value)** For example, to set the minimum PIN length to 10 characters, the Partition SO should set the value of this policy to 246: **256 - 10 = 246** The reason for this inversion is that a policy can only be set to a value equal to or lower than the value set by its capability. If the absolute minimum PIN length was set to 8, the Partition SO would be able to set the preferred minimum to 2, a less-secure policy. The Partition SO may only change the minimum PIN length to increase security by forcing stronger passwords. Default: **248**
26	Maximum PIN length	Maximum PIN length	The absolute maximum length for a partition login PIN is 255 characters. The effective maximum may be changed by the Partition SO, and must always be greater than the value of the minimum PIN length, determined by the formula in the description of policy 25 (above). Default: **255**
28	Enable Key Management Functions	Allow Key Management Functions	The Partition SO can disable access to any key management functions by the user - all users become Crypto Users (the restricted-capability user) even if logged in as Crypto Officer. Default: **ON** **Destructive: OFF-to-ON**
29	Enable RSA signing without confirmation	Perform RSA signing without confirmation	The HSM can perform an internal verification (confirmation) of a signing operation to validate the signature. This confirmation is disabled by default because it has a performance impact on signature operations. Default: **ON** **Destructive: OFF-to-ON**
31	Enable private key unmasking	Allow private key unmasking	Remove encryption with AES 256-bit key from private key Default: **ON**
32	Enable secret key unmasking	Allow secret key unmasking	Remove encryption with AES 256-bit key from secret key Default: **ON**

#	Partition Capability	Partition Policy	Description
33	Enable RSA PKCS mechanism	Allow RSA PKCS mechanism	Default: **ON** **Destructive: OFF-to-ON**
34	Enable CBC-PAD (un)wrap keys of any size	Allow CBC-PAD (un)wrap keys of any size	Default: **ON** **Destructive: OFF-to-ON**
37	Enable Secure Trusted Channel	Force Secure Trusted Channel	Secure Trusted Channel is a Network HSM feature, and has no function on SafeNet Luna PCIe HSM. Thales Group does not recommend turning this policy on at any time. Default: **OFF** **Destructive: ON-to-OFF**
39	Enable Start/End Date Attributes	Allow Start/End Date Attributes	If enabled, the Partition SO can turn this policy on to enforce CKA_START_DATE/CKA_END_DATE attributes for the partition. With the policy turned off, these attributes can be set, but their values will be ignored. Default: **OFF** **Destructive: ON-to-OFF**

Policy Templates

A policy template is a file containing a set of preferred HSM or partition policy settings, used to initialize HSMs/partitions with those settings. You can use the same file to initialize multiple HSMs or partitions, rather than changing policies manually after initialization. This can save time and effort when initializing HSMs or partitions that are to function as an HA group, or must comply with your company's overall security strategy. Templates enable scalable policy management and simplify future audit and compliance requirements.

> **NOTE** This feature has software and/or firmware dependencies. See "Version Dependencies by Feature" on page 317 for more information.

You can create a policy template file from an initialized or uninitialized HSM/partition, and edit it using a standard text editor. Partition policy templates have further customization options.

Policy templates cannot be used to alter settings for an initialized HSM or partition. Once an HSM or partition has been initialized, the SO must use **hsm changehsmpolicy** or **partition changepolicy** in LunaCM to change individual policy values. To zeroize the HSM and reset the policies to their default values, use **hsm factoryreset** in LunaCM on the admin partition.

To zeroize the HSM and keep the current policy settings, use **hsm zeroize** in LunaCM on the admin partition.

This section provides instructions for the following procedures, and some general guidelines and restrictions:

> "Creating a Policy Template" on the next page

> "Editing a Policy Template" below
> "Guidelines and Restrictions" on page 96
> "Applying a Policy Template" on page 97

Creating a Policy Template

The following procedures describe how to create a policy template for an HSM or partition.

To create an HSM policy template:

1. Launch LunaCM and set the active slot to the Admin partition. If you are creating a template from an initialized HSM, you must log in as HSM SO.

   lunacm:>**slot set slot** <admin_slotnum>

   lunacm:>**role login -name so**

2. Create the HSM policy template file with an original filename. Specify the path to the location where you wish to save the template. No file extension is required. If a template file with the same name exists in the specified directory, it is overwritten.

   lunacm:>**hsm showpolicies -exporttemplate** <filepath/filename>

   ```
   lunacm:>hsm showpolicies -templatefile /usr/safenet/lunaclient/templates/HSMPT

   HSM policies for HSM: myPCIeHSM written to /usr/safenet/lunaclient/templates/HSMPT

   Command Result : No Error
   ```

3. Customize the template file with a standard text editor (see "Editing a Policy Template" below).

To create a partition policy template:

1. Launch LunaCM and set the active slot to the partition. If you are creating a template from an initialized partition, you must log in as Partition SO.

   lunacm:>**slot set slot** <slotnum>

   lunacm:>**role login -name po**

2. Create the partition policy template file. Specify an existing save directory and the desired filename. No file extension is required. If a template file with the same name exists in the specified directory, it is overwritten.

   lunacm:>**partition showpolicies -exporttemplate** <filepath/filename>

   ```
   lunacm:> partition showpolicies -exporttemplate /usr/safenet/lunaclient/templates/ParPT

   Partition policies for Partition: myPartition1 written to
   /usr/safenet/lunaclient/templates/ParPT

   Command Result : No Error
   ```

Editing a Policy Template

Use a standard text editor to manually edit policy templates for custom configurations. This section provides template examples and customization guidelines.

HSM Policy Template Example

This example shows the contents of an HSM policy template created using the factory default policy settings. Use a standard text editor to change the policy values (0=OFF, 1=ON, or the desired value 0-255). You cannot edit the destructiveness of HSM policies. See "HSM Capabilities and Policies" on page 82 for more information.

If you export a policy template from an uninitialized HSM, the **Sourced from HSM** header field remains blank. This field is informational and you can still apply the template.

The **Policy Description** field is included in the template for user readability only. Policies are verified by the number in the **Policy ID** field.

```
# Policy template FW Version 7.1.0
# Field format - Policy ID:Policy Description:Policy Value
# Sourced from HSM: myLunaHSM, SN: 66331


6:"Allow masking":0
7:"Allow cloning":1
12:"Allow non-FIPS algorithms":1
15:"SO can reset partition PIN":0
16:"Allow network replication":1
21:"Force user PIN change after set/reset":1
22:"Allow offboard storage":1
23:"Allow partition groups":0
25:"Allow remote PED usage":0
30:"Allow unmasking":1
33:"Current maximum number of partitions":100
35:"Force Single Domain":0
36:"Allow Unified PED Key":0
37:"Allow MofN":0
38:"Allow small form factor backup/restore":0
39:"Allow Secure Trusted Channel":0
40:"Decommission on tamper":0
42:"Allow partition re-initialize":0
43:"Allow low level math acceleration":0
46:"Disable Decommission":1
47:"Allow Tunnel Slot":0
48:"Do Controlled Tamper Recovery":1
```

Partition Policy Template Example

This example shows the contents of a partition policy template created using the factory default policy settings. Use a standard text editor to change the policy and/or destructiveness values (0=OFF, 1=ON, or the desired value 0-255).

Partition policy template entries have two additional fields: **Off to on destructive** and **On to off destructive** (see example below). Change these values to **0** or **1** to determine whether cryptographic objects on the partition should be deleted when this policy is changed in the future. Policies that lower the security level of the objects stored on the partition are normally destructive, but it may be useful to customize this behavior for your own security strategy. See "Partition Capabilities and Policies" on page 87 for more information.

> **CAUTION!** Setting policy destructiveness to **0** (OFF) makes partitions less secure. Use this feature only if your security strategy demands it.

If you export a policy template from an uninitialized partition, the **Sourced from partition** header field remains blank. This field is informational and you can still apply the template.

The **Policy Description** field is included in the template for user readability only. Policies are verified by the number in the **Policy ID** field.

```
# Policy template FW Version 7.1.0
# Field format - Policy ID:Policy Description:Policy Value:Off to on destructive:On to off
destructive
# Sourced from partition: myPartition1, SN: 154438865290


0:"Allow private key cloning":1:1:0
1:"Allow private key wrapping":0:1:0
2:"Allow private key unwrapping":1:0:0
3:"Allow private key masking":0:1:0
4:"Allow secret key cloning":1:1:0
5:"Allow secret key wrapping":1:1:0
6:"Allow secret key unwrapping":1:0:0
7:"Allow secret key masking":0:1:0
10:"Allow multipurpose keys":1:1:0
11:"Allow changing key attributes":1:1:0
15:"Ignore failed challenge responses":1:1:0
16:"Operate without RSA blinding":1:1:0
17:"Allow signing with non-local keys":1:0:0
18:"Allow raw RSA operations":1:1:0
20:"Max failed user logins allowed":10:0:0
21:"Allow high availability recovery":1:0:0
22:"Allow activation":0:0:0
23:"Allow auto-activation":0:0:0
25:"Minimum pin length (inverted  255 - min)":248:0:0
26:"Maximum pin length":255:0:0
28:"Allow Key Management Functions":1:1:0
29:"Perform RSA signing without confirmation":1:1:0
31:"Allow private key unmasking":1:0:0
32:"Allow secret key unmasking":1:0:0
33:"Allow RSA PKCS mechanism":1:1:0
34:"Allow CBC-PAD (un)wrap keys of any size":1:1:0
39:"Allow Start/End Date Attributes":0:1:0
```

Guidelines and Restrictions

When creating, applying, or editing policy templates:

- You can remove a policy from the template by adding **#** at the beginning of the line or deleting the line entirely. When you apply the template, the partition will use the default values for that policy.
- Partition policy templates from older Luna versions (6.x or earlier) cannot be applied to Luna 7.x partitions.
- This version of the partition policy template feature is available on Luna 7.x user partitions only. When the active slot is set to a Luna 6.x partition or the Admin partition, the **-exporttemplate** option is not available. To create an HSM policy template from the Admin partition, use **hsm showpolicies -exporttemplate**.
- The following restrictions apply when configuring partitions for Cloning or Key Export (see "Keys In Hardware vs. Private Key Export" on page 172 for more information):
  - **Partition policy 0: Allow private key cloning** and **partition policy 1: Allow private key wrapping** can never be set to **1** (ON) at the same time. Initialization fails if the template contains a value of **1** for both policies.
  - **Partition policy 1: Allow private key wrapping** must always have **Off-to-on** destructiveness set to **1** (ON). Initialization fails if the template contains a value of **0** in this field.

> You may not use invalid policy values (outside the acceptable range), or values that conflict with your HSM or partition's capabilities. For example, **HSM capability 6: Enable Masking** is always **Disallowed**, so you cannot set the corresponding HSM policy to **1**. If you attempt to initialize an HSM or partition with a template containing invalid policy values, an error is returned and initialization fails:

```
lunacm:>hsm init -label myPartition1 -force -applytemplate ParPTbadvalue


The following values from the PPT are not compatible with the current hsm capabilities:

        3: Allow private key masking: 1 (Capability: 0)
        7: Allow secret key masking: 1 (Capability: 0)
       23: Allow auto-activation: 1 (Capability: 0)
       36: Allow Fast-Path: 1 (Capability: 0)

No initialization was performed.
Error:  'hsm init' failed. (C0000102 : RC_DATA_INVALID)

Command Result : 65535 (Luna Shell execution)
```

> If you include policies that are incompatible with the current HSM's firmware, initialization fails:

```
lunacm:>partition init -label myPartition2 -force -applytemplate ParPTunsupported


The following policies are not supported. Unsupported values will be ignored.

        9: Unsupported policy

Error:  'hsm init' failed. (C0000102 : RC_DATA_INVALID)


Command Result : 65535 (Luna Shell execution)
```

Applying a Policy Template

The following procedures describe how to apply HSM and partition policy templates.

To apply a policy template to a new HSM:

1. Ensure that the template file is saved on the workstation hosting the destination HSM.
2. Launch LunaCM and initialize the destination HSM using the policy template file. If the template file is not in the same directory as LunaCM, include the correct filepath.

   lunacm:>**hsm init -label** <label> **-applytemplate** <filepath/filename>
3. Verify that the template has been applied correctly by checking the partition's policy settings.

   lunacm:>**hsm showpolicies**

To apply a policy template to a new partition:

1. Ensure that the template file is saved on the client workstation.
2. Launch LunaCM, set the active slot to the new partition, and initialize the partition using the policy template file. If the template file is not in the same directory as LunaCM, include the correct filepath.

   lunacm:>**slot set slot** <slotnum>

lunacm:>**partition init -label** <label> **-applytemplate** <filepath/filename>

3. Verify that the template has been applied correctly by checking the partition's policy settings.

   lunacm:>**partition showpolicies -verbose**

CHAPTER 4: Configuration File Summary

Many aspects of SafeNet Luna HSM configuration and operation are controlled or adjusted by the Chrystoki.conf file (Linux/UNIX) or Crystoki.ini file (Windows). The examples in the table below are from a Windows crystoki.ini file.

The configuration file is organized into named sections, under which related configuration-affecting entries might appear. A basic configuration file is always present in the SafeNet Luna Client folder, installed by the SafeNet Luna Client installer, with default values assigned to the populated entries. In addition to the most basic sections and entries, some additional sections and entries can be included at installation time, if you select more than the minimal installation options for your HSM model(s).

In addition, new entries can be added, or existing entries can be adjusted by actions that you perform in SafeNet tools like LunaCM and vtl.

Finally, some sections or entries can be added or adjusted by manual editing of the Chrystoki.conf /Crystoki.ini file.

If you install SafeNet Luna Client where a previous version was installed, then the existing configuration file is saved and the new file adds to the existing content if appropriate. That is, if you have a SafeNet Luna HSM setup, already configured and tweaked to your satisfaction, those settings are preserved when you update to newer SafeNet Luna Client.

The following table lists sections and settings that you are likely to encounter in normal use of SafeNet products. Not all are applicable to every SafeNet Luna HSM. Each setting is named, with default values, allowed range of values, description of the item/setting, and remarks about any interactions between the current setting and others that you might configure.

Where the range is a file path, <luna_client_dir> specifies the path to your SafeNet Luna HSM client installation.

Setting	Range (Default)	Description
[Chrystoki2]		
LibNT=	(<luna_client_dir>\cryptoki.dll)	Path to the Chrystoki2 library.
LibNT32=	(<luna_client_ dir>\win32\libCryptoki2.dll)	Path to the Chrystoki2 library on 32-bit Windows systems only.
[Luna]		

Setting	Range (Default)	Description
PEDTimeout1=	(100000) ms	Specifies the PED timeout time 1 - defines how long (in milliseconds) the HSM tries to detect if it can talk to the PED before starting the actual communication with it. If the PED is unreachable the HSM returns to the host a result code for the respective HSM command. The result code indicates that the PED is not connected. This timeout is intended to be small so that the user is informed quickly that the PED is not connected.

Setting	Range (Default)	Description
PEDTimeout2=	(200000) ms	Specifies the PED timeout time 2 - defines how long (in milliseconds) the firmware waits for the local PED to respond to PED commands. PED commands should not be confused with PED-related HSM commands. An HSM sends PED commands to the PED when processing PED-related HSM commands, such as LOGIN or PED_CONNECT. One PED-related HSM command can involve many PED commands being sent by the HSM to the PED (for example, the MofN related commands). If a local PED does not respond to the PED commands within the span of PEDTimeout2 the HSM returns an appropriate result code (such as PED_TIMEOUT) for the respective PED-related HSM command.
PEDTimeout3=	(20000) ms	Specifies the PED timeout time 3 - defines additional time (in milliseconds) the firmware must wait for the remote PED to respond to PED commands. That is, the actual time the firmware waits for a remote PED to respond is PEDTimeout2 + PEDTimeout3.

Setting	Range (Default)	Description
DefaultTimeOut=	(500000) ms	Sets the default timeout interval - defines how long (in milliseconds) the HSM driver in the host system waits for HSM commands to return a result code. If the result code is not returned in that time, the driver assumes that the HSM is stuck and halts it, with the DEVICE_ ERROR returned to all applications that use the HSM. Most HSM commands use this timeout. Very few exceptions exist, when a command's timeout is hard-coded in the Cryptoki library, or separate timeouts are specified in the Chrystoki.conf for certain classes of HSM commands.
CommandTimeoutPedSet=	(720000) ms	This is an exception to DefaultTimeout (above). It defines timeout (in milliseconds) for all PED-related HSM commands. This class of PED-related commands can take more time than the ordinary commands that subscribe to the DefaultTimeOut value. As a rule of thumb, CommandTimeOutPedSe t = DefaultTimeOut + PEDTimeout1 + PEDTimeout2 + PEDTimeout3.

Setting	Range (Default)	Description
KeypairGenTimeOut=	(2700000) ms	The amount of time (in milliseconds) the library allows for a Keypair generate operation to return a value. Due to the random component, large key sizes can take an arbitrarily long time to generate, and this setting keeps the attempts within reasonable bounds. The default is calculated as the best balance between the inconvenience of occasional very long waits and the inconvenience of restarting a keygen operation. You can change it to suit your situation.
CloningCommandTimeout=	(300000) ms	The amount of time (in milliseconds) the library allows for the HSM to respond to a cloning command.
DomainParamTimeout=	(5400000) ms	Timeout for Domain Parameter Generation.
[CardReader]		
RemoteCommand=	0 = false (1 = true)	This setting was used when debugging older SafeNet products. For modern products it is ignored.

Setting	Range (Default)	Description
LunaG5Slots=	(3)	Number of SafeNet Luna USB HSM slots reserved so that the library will check for connected devices. > Can be set to zero if you have no SafeNet Luna USB HSMs and wish to get rid of the reserved spaces in your slot list. > Can be set to any number, but is effectively limited by the number of external USB devices your host can support.
[RBS]		
HostName=	Any hostname or IP address (0.0.0.0)	The hostname or IP address that the RBS server will listen on. Default is 0.0.0.0 (any IP on the local host).
HostPort=	Any unassigned port(1792)	The port number used by the RBS server.
ClientAuthFile=	(<luna_client_ dir>\config\clientauth.dat)	The location of the RBS Client authentication file.
ServerCertFile=	(<luna_client_ dir>\cert\server\server.pem)	The location of the RBS Server certificate file.
ServerPrivKeyFile=	(<luna_client_ dir>\cert\server\serverkey.pem)	The location of the RBS Server certificate private key file.
ServerSSLConfigFile=	(<luna_client_dir>\openssl.cnf)	The location of the OpenSSL configuration file used by RBS Server or Client.

Setting	Range (Default)	Description
CmdProcessor=	(<luna_client_dir>\rbs_processor2.dll)	The location of the RBS library.
NetServer=	0 = false (1 = true)	If true (default), RBS acts as a Server. If false, RBS acts as a Client.
[LunaSA Client]		
SSLConfigFile=	(<luna_client_dir>\openssl.cnf)	Location of the OpenSSL configuration file.
ReceiveTimeout=	(20000) ms	Time in milliseconds before a receive timeout

Setting	Range (Default)	Description
TCPKeepAlive=	0 = false (1 = true)	TCPKeepAlive is a TCP stack option, available at the LunaClient, and at the SafeNet Luna Network HSM appliance. For SafeNet purposes, it is controlled via an entry in the Chrystoki.conf /crystoki.ini file on the LunaClient, and in an equivalent file on SafeNet Luna Network HSM. For SafeNet Luna HSM 6.1 and newer, a fresh client software installation includes an entry "TCPKeepAlive=1" in the "LunaSA Client" section of the configuration file Chrystoki.conf (Linux/UNIX) or crystoki.ini (Windows). Config files and certificates are normally preserved through an uninstall, unless you explicitly delete them. As such, if you update (install) LunaClient software where you previously had an older LunaClient that did not have a TCPKeepAlive entry, one is added and set to "1" (enabled), by default. In the case of update, if TCPKeepAlive is already defined in the configuration file, then your existing setting (enabled or disabled) is preserved. The settings at the appliance and the client are independent. This allows a level of assurance, in case (for

Setting	Range (Default)	Description
		example) a firewall setting blocks in one direction.
NetClient=	0 = false (1 = true)	If true, library will search for network slots
ServerCAFile=	(<luna_client_ dir>\cert\server\CAFile.pem)	Location, on the client, of the server certificate file (set by vtl or LunaCM command **clientconfig deploy**).
ClientCertFile=	(<luna_client_ dir>\cert\client\ClientNameCert.pem)	Location of the Client certificate file that is uploaded to SafeNet Luna Network HSM for NTLS (set by vtl or LunaCM command **clientconfig deploy**).
ClientPrivKeyFile=	(<luna_client_ dir>\cert\client\ClientNameKey.pem)	Location of the Client private key file (set by vtl or LunaCM command **clientconfig deploy**).
ServerName00=192.20.17.200 ServerPort00=1792 ServerName01= ServerPort01=		Entries embedded by vtl utility, when you run the command **vtl addserver** or the LunaCM command **clientconfig deploy**. Identifies the NTLS-linked SafeNet Luna Network HSM servers, and determines the order in which they are polled to create a slot list.

> **NOTE** The **[Presentation]** section is not created automatically. To change any of the following values, you must first create this section in the configuration file.

[Presentation]

Setting	Range (Default)	Description
ShowUserSlots=<slot>(<serialnumber>)	Comma-delimited list of <slotnumber>(<serialnumber>), like ShowUserSlots=1(351970018022), 2(351970018021),3(351970018020),....	Sets the starting slot for the identified partition. If one partition slot on an HSM is specified, then any that are not listed from that HSM are not displayed.
ShowAdminTokens=	0/(1)	Admin partitions of local SafeNet Luna PCIe HSMs are not visible/ (visible) in a slot listing
ShowEmptySlots=	(0)/1	When the number of partitions on an HSM is not at the limit, unused slots are shown/(not shown).
OneBaseSlotId=	(0)/1	Causes basic slot list to start at slot number 1 instead of (0).

[HAConfiguration]

Setting	Range (Default)	Description
HAOnly=	(0)/1	When set to 1, shows only the HA virtual slot to the client, and hides the physical partitions/slots that are members of the virtual slot. Setting HAOnly helps prevent synchronization problems among member partitions, by forcing all client actions to be directed against the virtual slot, and dealing with synch transparently. HAOnly also prevents the shifting of slot numbers in the slot list that could occur if a visible physical partition were to drop out, which could disrupt an application that identifies its client partitions by slot numbers.
reconnAtt=	(10)	Specifies how many reconnection attempts will be made, when a member drops from the group. A value of "-1" is infinite retries.
AutoReconnectInterval=	(60) s	Specifies the interval (in seconds) at which the library will attempt to reconnect with a missing member, until "reconnAtt" is reached, and attempts cease. The default value of 60 seconds is the lowest that is accepted.
[Misc]		
ToolsDir=	(<luna_client_dir>\)	The location of the LunaClient tools.

Setting	Range (Default)	Description
RSAKeyGenMechRemap=	(0)/1	Controls what happens on newer firmware, when calls are made to specific older mechanisms that are now discouraged due to weakness. When this item is set to 0, no re-mapping is performed. When the value is set to 1, the following re-mapping occurs if the HSM firmware permits: > PKCS Key Gen -> 186-3 Prime key gen > X9.31 Key Gen -> 186-3 Aux Prime key gen (see "Mechanism Remap for FIPS Compliance" on page 1)

Setting	Range (Default)	Description
RSAPre1863KeyGen MechRemap=	(0)/1	Controls what happens on older firmware, when specific newer mechanisms are called, that are not supported on the older firmware. When this item is set to 0, no re-mapping is performed. When the value is set to 1, the following re-mapping occurs if the HSM firmware permits: > 186-3 Prime key gen -> PKCS Key Gen > 186-3 Aux Prime key gen -> X9.31 Key Gen Intended for evaluation purposes, such as with existing integrations that require newer mechanisms, before you update to firmware that actually supports the more secure mechanisms. Be careful with this setting, which makes it appear you are getting a new, secure mechanism, when really you are getting an outdated, insecure mechanism. (see "Mechanism Remap for FIPS Compliance " on page 1)

Setting	Range (Default)	Description
ProtectedAuthenticationPathFlagStatus=	(0)/1/2	This flag specifies which role to check for challenge request status. Possible values include: > **0** (default): no challenge request > **1**: check for Crypto Officer challenge request > **2**: check for Crypto User challenge request Edited using the configurator tool.
CopyRSAPublicValuesFromPrivateTemplate	0/(1)	Controls whether the public exponent of an RSA key can be copied from the private key template, if the public key template does not already have a public exponent attribute set. Possible values include: > **0**: if no public exponent is provided in the public template, then an error is returned (expected behavior). > **1**(default): if no public exponent is provided in the public template, then the private exponent is copied from the private template to populate the public template. For PKCS#11 compliance, this should be set to "0".

Setting	Range (Default)	Description
FunctionBindLevel=	[0]/1/2	This flag determines what action to take if a function binding fails during a CryptokiConnect() operation. Possible values include: > **0**: fail if not all functions can be resolved (default) > **1**: do not fail but issue warning for each function not resolved > **2**: do not fail and do not issue warning (silent mode)
[Secure Trusted Channel]		
ClientTokenLib= (for 64-bit Windows systems)	For soft token: > <luna_client_dir>\softtoken.dll For hard token: > C:\Windows\System32\etoken.dll	Specifies the location of the token library on 64-bit Windows systems. This value must be correct in order to use a client token. For 32-bit systems, see the **ClientTokenLib32** entry below. By default, **ClientTokenLib** points to the location of the soft token library. If you are using a hard token, you must manually change this value to point to the hard token library for your operating system. The exact location of the hard token library may vary depending on your installer. The location provided here is the most common location used.

Setting	Range (Default)	Description
ClientTokenLib32= (for 32-bit Windows systems)	For soft token: > <luna_client_ dir>\win32\softtoken.dll For hard token: > C:\Windows\SysWOW64\etoken.dll	Specifies the location of the token library on 32-bit Windows systems. This entry appears on Windows only. For 64-bit systems, see the **ClientTokenLib** entry above. By default, **ClientTokenLib32** points to the location of the soft token library. If you are using a hard token, you must manually change this value to point to the hard token library for your operating system. The exact location of the hard token library may vary depending on your installer. The location provided here is the most common location used.

NOTE The **[Session]** section is not created automatically. To change any of the following values, you must first create this section in the configuration file.

[Session]

Setting	Range (Default)	Description
AutoCleanUpDisabled=	(0)/1	This flag determines whether AutoCleanUp runs, to close orphan sessions, on behalf of an application that neglects to close sessions. It is useful for SafeNet Luna PCIe HSMs (in the host that also runs the client application). AutoCleanUp runs during C_Finalize on the client. By contrast, the SafeNet Luna Network HSM has the active NTLS service that tracks and closes dangling sessions. Possible values include: > **0**: Run AutoCleanUp if your application leaks sessions and you cannot rewrite the application. > **1**: Disable AutoCleanUp if you have a SafeNet Luna PCIe HSM and your client application does proper housekeeping, or if your application is connecting via NTLS to a SafeNet Luna Network HSM.

NOTE The **[Toggles]** section is not created automatically. To change any of the following values, you must first create this section in the configuration file.

[Toggles]

Setting	Range (Default)	Description
legacy_memory_rep =	(0)/1	Controls the manner in which the HSM reports the available RAM space. Possible values include: > **0** (default): the public and private memory total/free values reported in the CK_TOKEN_INFO structure indicate the available flash memory for permanent (TOKEN) objects that are in either the public or private space respectively; this method is PKCS#11 compliant. > **1**: the public memory values indicate the total/free RAM memory; this non-standard legacy method was used by some customers to determine space available for session based objects, and must be explicitly selected in order to continue using the legacy method.

CHAPTER 5: Decommissioning, Zeroizing, Re-imaging, or Resetting an HSM to Factory Conditions

During the lifetime of a SafeNet Luna HSM, you might have cause to take the HSM out of service, and wish to perform actions to ensure that no trace of your sensitive material remains. Those events might include:

- Placing the unit into storage, perhaps as a spare
- Shipping to another location or business unit in your organization
- Shipping the unit back to Gemalto for repair/re-manufacture
- Removing the HSM permanently from operational use, for disposal at end-of-life

This chapter describes the available options in the following sections:

- "Zeroization" below
- "Decommissioning the HSM Card" on the next page
- "Resetting to Factory Condition" on page 119
- "Comparing Zeroize, Decommission, and Factory Reset" on page 119
- "End of Service and Disposal" on page 120
- "Comparison of Destruction/Denial Actions" on page 121
- "RMA and Shipping Back to Thales Group" on page 123

Zeroization

In the context of HSMs in general, the term "zeroize" means to erase all plaintext keys. Some HSMs keep all keys in plaintext within the HSM boundary. SafeNet Luna HSMs do not.

In the context of SafeNet Luna HSMs, keys at rest (keys or objects that are stored in the HSM) are encrypted. Keys are decrypted into a volatile working memory space inside the HSM only while they are being used. Items in volatile memory disappear when power is removed. The action that we loosely call "zeroizing", or clearing, erases volatile memory as well as destroying the key that encrypts stored objects.

Any temporarily decrypted keys are destroyed, and all customer keys on the HSM are immediately rendered inaccessible and unrecoverable whenever you:

- perform **hsm factoryreset**
- make too many bad login attempts on the SO account
- short the pins of the decommission header
- set a "destructive" HSM policy
- perform HSM firmware rollback

The KEK (key encryption key that encrypts all user objects, partition structure, cloning vectors, masking vectors, etc.) is destroyed by a zeroization (erasure) or decommission event. At that point, any objects or identities in the HSM become effectively random blobs of bits that can never be decoded.

> **NOTE** The next HSM power-up following a KEK zeroization automatically erases the contents of user storage, which were already an indecipherable blob without the original KEK. That is, any zeroizing event instantly makes encrypted objects unusable, and as soon as power is re-applied, the HSM immediately erases even the encrypted remains before it allows further use of the HSM.

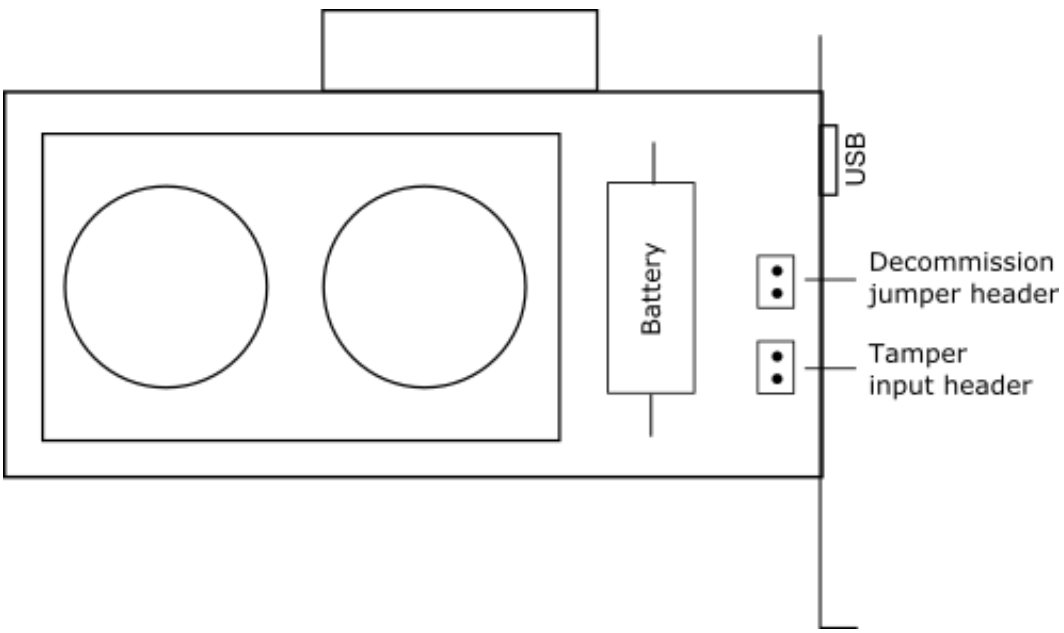
The HSM must now be re-initialized in order to use it again, and initialization overwrites the HSM with new user parameters. Everything is further encrypted with a new KEK unique to that HSM.

Keys not encrypted by the KEK are those that require exemption and are not involved in user identities or user objects:

- The Master Tamper Key, which enables tamper handling
- The Remote PED Vector, to allow Remote PED-mediated recovery from tamper or from Secure Transport Mode
- The hardware origin key that certifies the HSM hardware as having been built by Thales Group

Decommissioning the HSM Card

The SafeNet Luna PCIe HSM is equipped with a two-pin decommission jumper header, as illustrated below.



By default, short-circuiting the decommission jumper header decommissions the HSM. You can use the blade of a screwdriver, or other conductive tool to short-circuit the two pins of the decommission header, or you can connect a switch to the decommission header if desired. Power is not required to decommission the HSM, that is, you can decommission the HSM after removing it from the chassis.

When you decommission a SafeNet Luna PCIe HSM, the HSM is zeroized, all user accounts are deleted, and the HSM is returned to its factory state. Any firmware or partition upgrade packs installed on the HSM are retained.

You can also set **HSM Policy 40: Decommission on Tamper** to automatically decommission the HSM for selected tamper events. See "Tamper Events" on page 281 for details.

Disabling Decommissioning

You can disable the decommissioning feature if desired, by enabling **HSM Policy 46: Disable Decommission** (see "HSM Capabilities and Policies" on page 82). The primary reason for disabling decommissioning is to prevent the HSM from being automatically decommissioned due to loss of battery (see "Tamper Events" on page 281). If decommissioning is disabled, the SafeNet Luna PCIe HSM has an indefinite shelf life, as far as the battery is concerned.

To disable decommissioning

1. Launch LunaCM and log in as HSM SO.

   lunacm:>**role login -name so**

2. Enable **HSM Policy 46: Disable Decommission**:

   lunacm:> **hsm changehsmpolicy -policy 46 -value 1**

Resetting to Factory Condition

These instructions will allow you to restore your SafeNet Luna PCIe HSM to its original factory configuration. If you have performed firmware and software updates, those remain in place, and are not affected by this procedure. The reset commands affect contents and settings of the HSM. Reverting of software and firmware is outside their scope.

To reset the HSM to factory condition:

1. Login to the admin partition as HSM SO.

   role login -name so

2. Reset the HSM to factory settings.

   hsm factoryreset

Comparing Zeroize, Decommission, and Factory Reset

You can clear the contents of your HSM on demand, or the HSM may be cleared in response to an event. How this affects the contents and configuration of your HSM depends on whether the user partitions were deleted or whether the HSM was zeroized, decommissioned, or factory reset as detailed below:

Action	Command/Event	Description
Erase User Partitions	> Enable or disable a destructive HSM policy	Destroy/erase all user partitions, but do not zeroize the HSM. Policy 46 "Disable Decommission" is the exception in that it zeroizes the HSM and erases all user partitions if the policy is changed. To bring the HSM back into service, you need to: 1. Recreate the partitions 2. Reinitialize the partition roles
Zeroize	> Too many bad login attempts on the HSM SO account > Perform an HSM firmware rollback > lunacm:>**hsm zeroize**	Deletes all partitions and their contents, but retains the HSM configuration (audit role and configuration, policy settings). To bring the HSM back into service, you need to: 1. Reinitialize the HSM 2. Recreate the partitions 3. Reinitialize the partition roles
Decommission	> Press the decommission button on the rear of the appliance. > Enable **HSM Policy 40: Decommission on Tamper**, and tamper the HSM.	Deletes all partitions and their contents, the audit role, and the audit configuration. Retains the HSM policy settings. To bring the HSM back into service, you need to: 1. Reinitialize the HSM 2. Reinitialize the audit role and reconfigure auditing 3. Recreate the partitions 4. Reinitialize the partition roles
Factory Reset	lunacm:>**hsm factoryreset**	Deletes all partitions and their contents, and resets all roles and policy configurations to their factory default values. To bring the HSM back into service, you need to completely reconfigure the HSMas though it were new from the factory.

End of Service and Disposal

SafeNet Luna HSMs and appliances are deployed into a wide variety of markets and environments. Arranging for the eventual disposal of a SafeNet Luna HSM or appliance that is no longer needed can be a simple accounting task and a call to your local computer recycling service, or it can be a complex and rigorous set of procedures intended to protect very sensitive information.

Needs Can Differ

Some users of SafeNet Luna HSMs employ cryptographic keys and material that have a very short "shelf life". A relatively short time after the HSM is taken out of service, any objects that it contains are no longer relevant. The HSM could be disposed of, with no concern about any material that might remain in it.

The majority of our customers are concerned with their keys and objects that are stored on the HSM. It is important to them that those items never be exposed. The fact is that they are never exposed, but see below for explanations and actions that address the concerns of auditors who might be more accustomed to other ways of safeguarding HSM contents.

SafeNet Luna HSM Protects Your Keys and Objects

The design philosophy of our SafeNet Luna HSMs ensures that contents are safe from attackers. Unlike other HSM products on the market, SafeNet Luna HSMs never store sensitive objects, like cryptographic keys, unencrypted. Therefore, SafeNet Luna HSMs have no real need - other than perception or "optics" - to perform active erasure of HSM contents, in case of an attack or tamper event.

Instead, the basic state of a SafeNet Luna HSM is that any stored keys and objects are strongly encrypted. They are decrypted only for current use, and only into volatile memory within the HSM.

If power is removed from the HSM, or if the current session closes, the temporarily-decrypted objects instantly evaporate. The encrypted originals remain, but they are unusable by anyone who does not have the correct HSM keys to decrypt them.

How the HSM encryption keys protect your sensitive objects

In addition to encryption with the user specific access keys or passwords, all objects on the HSM are encrypted by the HSM's global key encryption key (KEK) and the HSM's unique Master Tamper Key (MTK).

If the HSM experiences a Decommission event (pressing of the small red button on back of SafeNet Luna Network HSM, or shorting of the pins of the decommission header on the HSM card, or removal of the battery while main power is not connected to a SafeNet Luna USB HSM) then the KEK is deleted.

If the HSM experiences a tamper event (physical intrusion, environmental excursion), then the MTK is destroyed.

Destruction of either of those keys instantly renders any objects in the HSM unusable by anyone. In the case of a Decommission event, when the HSM is next powered on, it requires initialization, which wipes even the encrypted remains of your former keys and objects.

We recognize that some organizations build their protocols around assumptions that apply to other suppliers' HSMs - where keys are stored unencrypted and must be actively erased in the event of an attack or removal from service. If your policies include that assumption, then you can re-initialize after Decommission - which actively erases the encrypted objects for which no decrypting key existed. For purposes of security, such an action is not required, but it can satisfy pre-existing protocols that presume a weakness not present in SafeNet Luna HSMs.

Comparison of Destruction/Denial Actions

Various operations on the SafeNet Luna PCIe HSM are intended to make HSM contents unavailable to potential intruders. The effect of those actions are summarized and contrasted in the following table, along with notes on how to recognize and how to recover from each scenario.

Scenario 1: MTK is destroyed, HSM is unavailable, but use/access can be recovered after reboot (See Note 1)

Scenario 2: KEK is destroyed (Real-Time Clock and NVRAM), HSM contents cannot be recovered without restore from backup See Note 2)

Event	Scen. 1	Scen. 2	How to discover (See Note 3)	How to recover
> Three bad SO login attempts > lunacm:> **hsm zeroize** > lunacm:> **hsm factoryreset** > Any change to a destructive policy > Firmware rollback (See Note 4)	NO	YES	> Log entry > "Partition Status -> Zeroized" in HSM info (from **hsm showinfo** on admin partition)	Restore HSM objects from Backup
Hardware tamper > Undervoltage or overvoltage during operation > Under-temperature or over-temperature during operation > Chassis interference (such as cover, fans, etc.) Software (command-initiated) tamper > lunacm:> **stm transport**	YES	NO	Parse logs for text like "tamper", "TVK was corrupted", or "Generating new TVK", indicating that a tamper event was logged. Example: `RTC: external tamper latched/ MTK: security function was zeroized on previous tamper event and has not been restored yet` Also, keywords in logs like: "HSM internal error", "device error"	Reboot [See Note 1]
Decommission > Short-circuiting the tamper header pins	NO	YES	Look for log entry like: RTC: tamper 2 signal/Zeroizing HSM after decommission...LOG(INFO): POWER-UP LOG DUMP END	Restore HSM objects from Backup

Note 1: MTK is an independent layer of encryption on HSM contents, to manage tamper and Secure Transport Mode. A destroyed MTK is recovered on next reboot. If MTK cannot be recovered, only restoring from backup onto a new or re-manufactured HSM can retrieve your keys and HSM data.

Note 2: KEK is an HSM-wide encryption layer that encrypts all HSM objects, excluding only MTK, RPK, a wrapping key, and a couple of keys used for legacy support. A destroyed KEK cannot be recovered. If the KEK is destroyed, only restoring from backup can retrieve your keys and HSM data.

Note 3: To check the health of a remote HSM, script a frequent login to the HSM host and execution of a subset of HSM commands. If a command fails, check the logs for an indication of the cause.

Note 4: These actions all create a situation where **hsm init** is required, or strongly recommended before the HSM is used again.

In addition, another event/action that has a destructive component is HSM initialization. See "HSM Initialization" on page 160.

RMA and Shipping Back to Thales Group

Although rare, it could happen that you need to ship a SafeNet appliance back to Thales Group.

Contact your Thales representative to obtain the Return Material Authorization (RMA) and instructions for packing and shipping.

You might wish (or your security policy might require you) to take maximum precaution with any contents in your HSM before it leaves your possession.

If so, there are two options available to secure the contents of the SafeNet Luna PCIe HSM before returning it to Thales Group:

> Decommission the HSM, forcibly clearing all HSM contents (see "Decommissioning the HSM Card" on page 118 for instructions).

> Set Secure Transport Mode on the HSM (see "Secure Transport Mode" on page 262 for instructions) and provide the verification string and random user string to your Thales Group representative by secure means. This will allow Thales Group to know if the HSM is tampered while in transit.

CHAPTER 6: High-Availability Groups

SafeNet Luna HSMs can provide scalability and redundancy for cryptographic applications that are critical to your organization. For applications that require continuous, uninterruptible uptime, the SafeNet Luna HSM Client allows you to combine application partitions on multiple HSMs into a single logical group, known as a High-Availability (HA) group.

An HA group allows your client application to access cryptographic services as long as one member HSM is functional and network-connected. This allows you to perform maintenance on any individual member without ever pausing your application, and provides redundancy in the case of individual failures. Cryptographic requests are distributed across all active group members, enabling a performance gain for each member added. Cryptographic objects are replicated across the entire group, so HA can also be used to keep a current, automatic, remote backup of the group contents.

HA functionality is handled by the SafeNet Luna HSM Client software. The individual partitions have no way to know they are configured in an HA group, so you can configure HA on a per-application basis. The way you group your HSMs depends on your circumstances and desired performance.

This chapter contains the following sections:

> "How HA Works" on page 126
> "Planning Your HA Group Deployment" on page 134
> "Setting Up an HA Group" on page 136
> "Managing Your HA Groups" on page 150
> "HA Troubleshooting" on page 158

Performance

For repetitive operations (for example, many signings using the same key), an HA group provides linear performance gains as group members are added. The best approach is to maintain an HA group at a size that best balances application server capability and the expected loads, with an additional unit providing capacity for bursts of traffic.

Load Balancing

Cryptographic requests sent to the HA group's virtual slot are load-balanced across all active members of the HA group. The load-balancing algorithm sends requests for cryptographic operations to the least busy partition in the HA group. This scheme accounts for operations of variable length, ensuring that queues are balanced even when some partitions are assigned very long operations. When an application requests a repeated set of operations, this method works. When the pattern is interrupted, however, the request type becomes relevant, as follows:

> Single-part (stateless) cryptographic operations are load-balanced.
> Multi-part (stateful) cryptographic operations are load-balanced.

- Multi-part (stateful) information retrieval requests are not load-balanced. In this case, the cost of distributing the requests to different HA group members is generally greater than the benefit. For this reason, multi-part information retrieval requests are all targeted at one member.
- Key management requests are not load-balanced. Operations affecting the state of stored keys (creation, deletion) are performed on a single HA member, and the result is then replicated to the rest of the HA group.

Key Replication

When an application creates a key on the virtual HA slot, the HA library automatically replicates the key across all group members before reporting back to the application. Keys are created on one member partition and replicated to the other members. If a member fails during this process, the HA group reattempts key replication to that member until it recovers, or failover attempts time out. Once the key exists on all active members of the HA group, a success code is returned to the application.

All key replication uses the Luna cloning protocol, which provides mutual authentication, confidentiality, and integrity for each object that is copied from one partition to another. Therefore, all HA group member partitions must be initialized with the same cloning domain.

Failover

When any active HA group member fails, a failover event occurs – the affected partition is dropped from the list of available HA group members, and all operations that were pending on the failed partition are transparently rescheduled on the remaining member partitions. The SafeNet Luna HSM Client continuously monitors the health of member partitions at two levels:

- network connectivity – disruption of the network connection causes a failover event after a 20-second timeout.
- command completion – any command that is not executed within 20 seconds causes a failover event.

As long as one HA group member remains functional, cryptographic service is maintained to an application no matter how many other group members fail.

Recovery

Recovery of a failed HA group member is designed to be automatic in as many cases as possible. You can configure your auto-recovery settings to require as much manual intervention as is convenient for you and your organization. In either an automated or manual recovery process, there is no need to restart your application. As part of the recovery process:

- Any cryptographic objects created while the member was offline are automatically replicated to the recovered partition.
- The recovered partition becomes available for its share of load-balanced cryptographic operations.

Automatic Recovery

With automatic recovery, the client library automatically performs periodic recovery attempts while a member is failed. The frequency of these checks is adjustable. Most customers enable auto-recovery in all configurations.

Manual Recovery
Simply run the client recovery command and the recovery logic inside the client makes a recovery attempt the next time the application uses the HSM. As part of recovery, any key material created while the member was offline is automatically replicated to the recovered unit.

Even if a manual recovery process is selected, the application does not need to be restarted.

Permanent Failure
Sometimes a failure of a device is permanent (for example, if the HSM is re-initialized). In this event, you only need to remove the failed unit and deploy a new member to the group. The running clients automatically resynchronize keys to the new member and start scheduling operations to it.

Standby Members

After you add member partitions to an HA group, you can designate some as standby members. Cryptographic objects are replicated on all members of the HA group, including standby members, but standby members do not perform any cryptographic operations unless all the active members go offline. In this event, all standby members are immediately promoted to active service, and operations are load-balanced across them. This provides an extra layer of assurance against a service blackout for your application.

How HA Works

This section provides detailed descriptions of the following aspects of HA functionality:

> "Performance" below
> "Load Balancing" on the next page
> "Key Replication" on page 128
> "Failover" on page 129
> "Recovery" on page 130
> "Standby Members" on page 131
> "Process Interaction" on page 132
> "Application Object Handles" on page 132
> "Example: Database Encryption" on page 133

Performance

For repetitive operations (for example, many signings using the same key), an HA group provides linear performance gains as group members are added. The best approach is to maintain an HA group at a size that best balances application server capability and the expected loads, with an additional unit providing capacity for bursts of traffic.

For best overall performance, keep all group members running near their individual performance ideal, about 30 simultaneous threads per HSM. If you assemble an HA group that is significantly larger than your server(s) can manage, you might not achieve full performance from all members. Gigabit Ethernet connections are recommended to maximize performance.

Performance is also affected by the kind of cryptographic operations being requested. For some operations, an HA group can actually hinder performance by requiring extra operations to replicate new key objects. For example, if the operation involves importing and unwrapping keys:

Using an HA group	Using an individual partition

1. Encryption (to wrap the key)
2. Decryption on one member partition (to unwrap the key)
3. Object creation on the same member partition (the unwrapped key is created and stored as a key object)
4. Key replication across the HA group:
  a. RSA 4096-bit operation is used to derive a shared secret between HSMs
  b. Encryption of the key on the original HA member using the shared secret
  c. Decryption of the key on each HA member using the shared secret
  d. Object creation on each HA member
5. Encryption (using the unwrapped key object to encrypt the data)
	1. Encryption (to wrap the key)
	2. Decryption (to unwrap the key)
	3. Object creation (the unwrapped key is created and stored as a key object)
	4. Encryption (using the unwrapped key object to encrypt the data)

In this case, the HA group must perform many more operations than an individual partition, most significantly the RSA-4096-bit operation and creating the additional objects. Those two operations are by far the most time-consuming on the list, and so this task would have much better performance on an individual partition.

The crucial HA performance consideration is whether the objects on the partitions are constant, or always being created and replaced. If tasks make use of already-existing objects, those objects exist on all HA group members; operations can be performed by different group members, boosting performance. If new objects are created, they must be replicated across the entire group, causing a performance loss.

> **NOTE** The way your application uses the **C_FindObjects** function to search for objects in a virtual HA slot can have a significant impact on your application performance (see "Application Object Handles" on page 132).

Load Balancing

Cryptographic requests sent to the HA group's virtual slot are load-balanced across all active members of the HA group. The load-balancing algorithm sends requests for cryptographic operations to the least busy partition in the HA group. This scheme accounts for operations of variable length, ensuring that queues are balanced even when some partitions are assigned very long operations. When an application requests a repeated set of operations, this method works. When the pattern is interrupted, however, the request type becomes relevant, as follows:

- Single-part (stateless) cryptographic operations are load-balanced.
- Multi-part (stateful) cryptographic operations are load-balanced.

- Multi-part (stateful) information retrieval requests are not load-balanced. In this case, the cost of distributing the requests to different HA group members is generally greater than the benefit. For this reason, multi-part information retrieval requests are all targeted at one member.
- Key management requests are not load-balanced. Operations affecting the state of stored keys (creation, deletion) are performed on a single HA member, and the result is then replicated to the rest of the HA group.

For example, when a member partition is signing and an asymmetric key generation request is issued, additional operations on that member are queued while the partition generates the key. In this case, the algorithm schedules more operations on other partitions in the HA group.

The load-balancing algorithm operates independently in each application process. Multiple processes on the same client or on different clients do not share information when scheduling operations. Some mixed-use cases might cause applications to use some partitions more than others (see "Planning Your HA Group Deployment" on page 134). If you increase key sizes, interleave other cryptographic operations, or if network latency increases, performance may drop for individual active members as they become busier.

> **NOTE** Partitions designated as standby members are not used to perform cryptographic operations, and are therefore not part of the load-balancing scheme (see "Standby Members" on page 131).

Network Topography

The network topography of the HA group is generally not important to the functioning of the group. As long as the client has a network path to each member, the HA logic will function. Different latencies between the client and each HA member cause a command scheduling bias towards the low-latency members. Commands scheduled on the long-latency devices have a longer overall latency associated with each command.

In this case, the command latency is a characteristic of the network. To achieve uniform load distribution, ensure that partitions in the group have similar network latency.

Key Replication

When an application creates a key on the virtual HA slot, the HA library automatically replicates the key across all group members before reporting back to the application. Keys are created on one member partition and replicated to the other members. If a member fails during this process, the HA group reattempts key replication to that member until it recovers, or failover attempts time out. Once the key exists on all active members of the HA group, a success code is returned to the application.

All key replication uses the Luna cloning protocol, which provides mutual authentication, confidentiality, and integrity for each object that is copied from one partition to another. Therefore, all HA group member partitions must be initialized with the same cloning domain.

The cloning protocol is invoked separately for each object to be cloned and the sequence of required calls must be issued by an authorized client library (residing on a client platform that has been authenticated to each of the partitions in the HA group). This ensures that the use of cloning function calls is controlled, and the protocol cannot be misused to permit the unauthorized transfer of objects to or from one of the partitions in the HA group.

Failover

When any active HA group member fails, a failover event occurs – the affected partition is dropped from the list of available HA group members, and all operations that were pending on the failed partition are transparently rescheduled on the remaining member partitions. The SafeNet Luna HSM Client continuously monitors the health of member partitions at two levels:

- network connectivity – disruption of the network connection causes a failover event after a 20-second timeout.
- command completion – any command that is not executed within 20 seconds causes a failover event.

> **NOTE** Most commands are completed within milliseconds. Some can take longer, either because the command itself is time-consuming (for example, key generation), or because the HSM is under extreme load. The HSM automatically sends a "heartbeat" signal every two seconds for commands that are pending or in progress. The client extends the 20-second timeout whenever it receives a heartbeat, preventing false failover events.

When an HA group member fails, the HA group status (see "hagroup listgroups" on page 1 in the *LunaCM Command Reference Guide*) reports a device error for the failed member. The client tries to reconnect the failed member at a minimum retry rate of once every 60 seconds, for the specified number of times (see "Recovery" on the next page).

When a failover occurs, the application experiences a latency stall on the commands in process on the failing unit, but otherwise there is no impact on the transaction flow. The scheduling algorithm described in "Load Balancing" on page 127 automatically minimizes the number of commands that stall on a failing unit during the 20-second timeout.

As long as one HA group member remains functional, cryptographic service is maintained no matter how many other group members fail. As described in "Recovery" on the next page, members can be returned to service without restarting the application.

Mid-operation failures

Any operation that fails mid-point needs to be re-sent from the calling application. The entire operation returns a failure (CKR_DEVICE_ERROR). This is more likely to happen in a multi-part operation, but a failure could conceivably happen during a single atomic operation as well.

For example, multi-part operations could be block encryption/decryption or any other command where the previous state of the HSM is critical to the processing of the next command. These operations must be re-sent, since the HA group does not synchronize partitions' internal memory state, only the stored key material.

> **NOTE** You must ensure that your applications can deal with the rare possibility of a mid-operation failure, by re-issuing the affected commands.

Possible Causes of Failure

In most cases, a failure is a brief service interruption, like a system reboot. These temporary interruptions are easily dealt with by the failover and auto-recovery functions. In some cases, additional actions may be required before auto-recovery can take place. For example, if a partition becomes deactivated, it must be reactivated by the Crypto Officer (see "Activation and Auto-Activation on PED-Authenticated Partitions" on page 178). Some permanent failures may require manual recovery (see "Recovery" below). Possible failure events include:

> **HSM-side failures**
  - HSM card failure
  - HSM re-initialization
  - HSM reboot
  - Deactivated partition
> **Client-side failures**
  - Client workstation power failure
  - Client workstation reboot

Recovery

Recovery of a failed HA group member is designed to be automatic in as many cases as possible. You can configure your auto-recovery settings to require as much manual intervention as is convenient for you and your organization. In either an automated or manual recovery process, there is no need to restart your application. As part of the recovery process:

> Any cryptographic objects created while the member was offline are automatically replicated to the recovered partition.

> The recovered partition becomes available for its share of load-balanced cryptographic operations.

Auto-recovery

When auto-recovery is enabled, SafeNet Luna HSM Client performs periodic recovery attempts when it detects a member failure. You can adjust the frequency (maximum once per minute) and the total number of retries (no limit). If the failed partition is not recovered within the scheduled number of retries, it remains a member of the HA group, but the client will no longer attempt to recover it. You must then address whatever equipment or network issue caused the failure, and execute a manual recovery of the member partition.

With each recovery attempt, a single application thread experiences a slight latency delay of a few hundred milliseconds while the client uses the thread to recover the failed member partition.

There are two HA auto-recovery modes:

> **activeBasic** – uses a separate, non-session-based Active Recovery Thread to perform background checks of HA member availability, recover failed members, and synchronize the contents of recovered members with the rest of the group. It does not restore existing sessions if all members fail simultaneously and are recovered.

> **activeEnhanced** – works the same as activeBasic, but restores existing sessions and login states if all members fail and are recovered.

HA auto-recovery is disabled by default. It is automatically enabled when you set the recovery retry count (see "Configuring HA Auto-Recovery" on page 144). Thales Group recommends enabling auto-recovery in all configurations.

> **NOTE** If a member partition loses Activation when it fails (it remains offline for more than two hours) you must present the black Crypto Officer PED key to re-cache the PED secret before the member can be recovered.

Manual Recovery

When auto-recovery is disabled, or fails to recover the partition within the scheduled number of retries, you must execute a manual recovery in LunaCM. Even if you use manual recovery, you do not need to restart your application. When you execute the recovery command, the client makes a recovery attempt the next time the application uses the group member (see "Manually Recovering a Failed HA Group Member" on page 154).

Even with auto-recovery enabled and configured for a large number of retries, there are some rare occasions where a manual recovery may be necessary (for example, when a member partition and the client application fail at the same time).

> **CAUTION!** Never attempt a manual recovery while the application is running and auto-recovery is enabled. This can cause multiple concurrent recovery processes, resulting in errors and possible key corruption.

Failure of All Group Members

If all members of an HA group fail (and no standby members are configured), all logged-in sessions are lost, and operations that were active when the last member failed are terminated. If you have set the HA auto-recovery mode to activeEnhanced, all sessions will be restarted when one or more members are recovered, and normal operations will resume. Otherwise, you must restart the client application once the group members have been recovered.

Permanent Failures

Sometimes an HSM failure is permanent (from the perspective of the HA group). For example, if the HSM is re-initialized, the member partition is erased and must be recreated. In this case, you can decide to recreate the original member or deploy a new member to the group. The client automatically replicates cryptographic objects to the new member and begins assigning operations to it (see "Replacing an HA Group Member" on page 155).

Standby Members

After you add member partitions to an HA group, you can designate some as standby members. Cryptographic objects are replicated on all members of the HA group, including standby members, but standby members do not perform any cryptographic operations unless all the active members go offline. In this event, all standby members are immediately promoted to active service, and operations are load-balanced across them. This provides an extra layer of assurance against a service blackout for your application. See "Planning Your HA Group Deployment" on page 134 for guidelines on how to make the most of your standby members.

Since standby members replicate keys but do not perform operations, they can also serve as an automatic backup partition for the cryptographic objects on the HA group. The contents of standby partitions are always kept up-to-date, so it is not possible to keep multiple backups using an HA group (see "Planning Your HA Group Deployment" on page 134).

Process Interaction

At the lowest communication level, the transport protocol (TCP) maintains communication between the client and the appliance (whether HA is involved or not). For HA groups involving member partitions on SafeNet Luna PCIe HSM, the protocol timeout is 10 seconds. This means:

- In a period of no activity by client or appliance, the appliance's TCP will wonder if the client is still there, and send a packet after 10 seconds of silence.
- If that packet is acknowledged, the 10-second TCP timer restarts, and the cycle repeats indefinitely.
- If the packet is not acknowledged, TCP sends another every 10 seconds. If there is no response after 2 minutes, the connection is considered dead, and higher levels are alerted to perform their cleanup.

Above that level, the NTLS/STC layer provides the connection security and some other services. Any time a client sends a request for a cryptographic operation, the HSM on the appliance begins working on that operation.

While the HSM processes the request, appliance-side NTLS/STC sends a "keep-alive" ping every 2 seconds, until the HSM completes the request. NTLS/STC does not perform any interpretation of the ping, but simply keeps the TCP layer active. If your client application requests a lengthy operation (for example, an 8192-bit keygen), the random-number-generation portion of that operation could take minutes, during which the HSM would legitimately be sending nothing back to the client. The NTLS ping ensures that the connection remains alive during long pauses.

Application Object Handles

Application developers should be aware that the PKCS #11 object handle model is fully virtualized when using an HA slot. The application must not assume fixed handle numbers across instances of an application. A handle's value remains consistent for the life of a process; but it might be a different value the next time the application is executed.

When you use an HA slot with your applications, the client behaves as follows when interacting with the application:

1. Intercept the call from the application.
2. Translate virtual object handles to physical object handles using the mappings specified by the virtual object table. The virtual object table is created and updated for the current session only, and only contains of list of the objects accessed in the current session.
3. Launch any required actions on the appropriate HSM or partition.
4. Receive the result from the HSM or partition and forward the result to your application,
5. Propagate any changes in objects on the physical HSM that performed the action to all of the other members of the HA group.

Virtual slots and virtual objects

When an application uses a non-HA physical slot, it addresses all objects in the slot by their physical object handles. When an application uses an HA slot, however, a virtual layer of abstraction overlays the underlying physical slots that make up the HA group, and the HA group is presented to the application as a virtual slot. This virtual slot contains virtual objects that have virtual object handles. The object handles in an HA slot are virtualized since the object handles on each of the underlying physical slots might be different from slot to slot. Furthermore, the physical object handles could change if a member of the HA group drops out (fails or loses communication) and is replaced.

The virtual object table

HA slots use a virtual object table to map the virtual objects in the virtual HA slot to the real objects in the physical slots that make up the HA group. The HA client builds a virtual object table for each application that loads the library. The table is ephemeral, and only exists for the current session. It is created and updated, if necessary, each time an application makes a request to access an object. To maximize performance and efficiency, the table only contains a list of the objects accessed in the current session. For example, the first time an application accesses an object after application start up, the table is created, a look up is performed to map the virtual object to its underlying physical objects, and an entry for the object is added to the table. For each subsequent request for that object, the data in the table is used and no look up is required. If the application then accesses a different object that is not listed in the table, a new look up is performed and the table is updated to add an entry for the new object.

C_FindObjects behavior and application performance

Since the client must perform a lookup to create the virtual object table, the way you use the C_FindObjects function can have a significant impact on the performance of your applications. For example, if you use the C_FindObjects function to ask for specific attributes, the client only needs to update the table to include the requested objects. If, however, you use the C_FindObjects function to find all objects, the client queries each HSM/partition in the group, for each object, to create the table. This can take a significant amount of time if the slot contains a large number of objects, or if the HA group includes many members.

To mitigate performance degradation when using the C_FindObjects function to list the objects on an HA slot, we recommend that you structure your applications to search by description, handles, or other attributes, rather than searching for all objects. Doing so minimizes the number of objects returned and the time required to create or update the table. If your application must find all objects, we recommend that you add the C_FindObjects all function call to the beginning of your application so that the table is built on application start up, so that the table is available to the application for all subsequent C_FindObjects function calls.

Example: Database Encryption

This section walks through a sample use case of some of the HA logic with a specific application – a transparent database encryption.

Typical Database Encryption Key Architecture

Database engines typically use a two-layered key architecture. At the top layer is a master encryption key that is the root of data protection. Losing this key is equivalent to losing the database, so it obviously needs to be highly durable. At the second layer are table keys used to protect table-spaces and/or columns. These table keys are stored with the database as blobs encrypted by the master encryption key (MEK). This architecture maps to the following operations on the HSM:

1. Initial generation of master key for each database.
2. Generation and encryption of table keys with the master key.
3. Decryption of table keys when the database needs to access encrypted elements.
4. Generation of new master keys during a re-key and then re-encrypting all table keys with it.
5. Generation and encryption of new table keys for storage in the database (often done in a software module).

The HSM is not involved in the use of table keys. Instead it provides the strong protection of the MEK which is used to protect the table keys. Users must follow backup procedures to ensure their MEK is as durable as the database itself ("Backup and Restore" on page 34).

HSM High Availability with Database Encryption

When the HSMs are configured as an HA group, the database's master key is automatically and transparently replicated to all the members when the key is created or re-keyed. If an HSM group member was offline or fails during the replication, it does not immediately receive a copy of the key. Instead the HA group proceeds after replicating to all of the active members. Once a member is re-joined to the group the HSM client automatically replicates the new master keys to the recovered member.

Before every re-key event, the user must ensure the HA group has sufficient redundancy. A re-key will succeed as long as one HA group member exists, but proceeding with too few HSMs will result in an availability risk. For example, proceeding with only one HSM means the new master key will be at risk since it exists only on a single HSM. Even with sufficient redundancy, Thales Group recommends maintaining an offline backup of a database's master key.

HSM Load Balancing with Database Encryption

While a database is up and running, the master key exists on all members in the HA group. Requests to encrypt or decrypt table keys are distributed across the entire group. The load-balancing feature is able to deliver improved performance and scalability when the database requires a large number of accesses to the table keys. Most deployments will not need much load balancing as the typical database deployment results in a small number of table keys.

While the table keys are re-keyed, new keys are generated in the HSM and encrypted for storage in the database. Within an HA group, these keys are generated on a single member and then replicated to the entire HA group, even though they exist on the HSM for only a moment. These events are infrequent enough that this extra replication has minimal impact.

Planning Your HA Group Deployment

This section describes important considerations and constraints to keep in mind as you plan your High-Availability (HA) group deployment. The benefits of HA are described in detail in "How HA Works" on page 126. There are several sample configurations described in this section that take advantage of different HA features. Depending on your organization's security needs, you might choose one of these configurations, or your own variation.

> "HSM and Partition Prerequisites" below
> "Sample Configuration" on the next page
  • "Performance and Load Balancing" on the next page

HSM and Partition Prerequisites

The HSM partitions you plan to use in an HA group must meet the following prerequisites before you can use them in an HA group.

Compatible HSM Firmware Versions

All HSMs in an HA group must have the same firmware version installed.

Common Cloning Domain

All key replication in an HA group uses the Luna cloning protocol, which provides mutual authentication, confidentiality, and integrity for each object that is copied from one partition to another. Therefore, all HA group member partitions must be initialized with the same cloning domain. If you are planning to combine already-existing partitions into an HA group, you must first re-initialize them using the same domain string or red PED key.

Common Crypto Officer Credentials

An HA group essentially allows you to log in to all its member partitions simultaneously, using a single credential. Password-authenticated partitions must all be initialized with the same Crypto Officer password. PED-authenticated partitions must all be initialized with the same black Crypto Officer PED key and activated with the same CO challenge password.

It is not possible to create an HA group made up of both password- and PED-authenticated partitions.

Common HSM/Partition Policies (FIPS Mode)

Generally, all HSMs/partitions used in an HA group must have the same policy configuration, especially FIPS mode. Do not attempt to use an HA group combining HSMs with FIPS mode on and others with FIPS mode off.

Sample Configuration

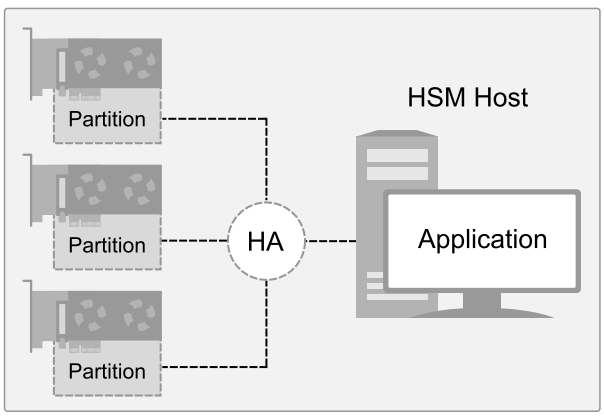
Your ideal HA group configuration depends on the number of HSMs you have available and the purpose of your application(s).

Performance and Load Balancing

If your application is designed to perform many cryptographic operations as quickly as possible, using keys or other objects that do not change often, you can create a large HA group using partitions on many HSMs. This deployment uses load balancing to provide linear performance gains for each HSM added to the group.

For example: your application uses keys stored on the HSM to perform many encrypt/decrypt or sign/verify operations. You want to minimize transaction latency by providing enough HSMs to handle capacity.

The SafeNet Luna HSM Client allows HA groups with up to 32 member partitions. The best approach in this example is to add enough group members to handle the usual number of operations, plus enough extra members to handle periods of high demand.



Setting Up an HA Group

Use LunaCM to create an HA group from partitions assigned to your client. This procedure is completed by the Crypto Officer. Ensure that you have met all necessary prerequisites before proceeding with group creation. For a detailed description of HA functionality, see "How HA Works" on page 126.

> **NOTE** Your LunaCM instance needs to update the **Chrystoki.conf** (Linux/UNIX) or **crystoki.ini** file (Windows) when setting up or reconfiguring HA. Ensure that you have Administrator privileges on the client workstation.

Prerequisites

HA groups are set up in LunaCM by the Crypto Officer. Before the CO can perform this setup, however, all HSMs and member partitions must meet the following prerequisites, completed by the HSM and Partition Security Officers.

HSMs

The HSM SO must ensure that all HSMs containing HA group member partitions meet the following prerequisites:

- All HSMs must be the same hardware type (a mix of Network and PCIe HSMs is not supported) and use the same authentication method (Password/PED).
- All HSMs must have the same firmware version installed.
- All must be installed in the same host server that will create the HA group.
- HSM policies **7: Allow Cloning** and **16: Allow Network Replication** must be set to **1** (see "Set the HSM Policies" on page 1 in the *Configuration Guide*).
- HSM policies must be consistent across all HSMs, particularly **12: Allow non-FIPS algorithms**. Do not attempt to use an HA group combining HSMs with FIPS mode on and others with FIPS mode off.

Partitions

The Partition SO must ensure that all partitions in an HA group meet the following prerequisites:

- All partitions must be visible in LunaCM on the host workstation.
- All partitions must be initialized with the same cloning domain:
  - Password-authenticated partitions must share the same domain string.
  - PED-authenticated partitions must share the same red domain PED key.
- Partition policies **0: Allow private key cloning** and **4: Allow secret key cloning** must be set to **1** on all partitions.
- Partition policies must be consistent across all member partitions.
- The Crypto Officer role on each partition must be initialized with the same CO credential (password or black PED key).

> PED-authenticated partitions must have partition policies **22: Allow activation** and **23: Allow auto-activation** set to **1**. All partitions must be activated and have auto-activation enabled, so that they can retain their login state after failure/recovery. Each partition must have the same activation challenge secret set (see "Activation and Auto-Activation on PED-Authenticated Partitions" on page 178)

> **NOTE** If HSM policy **21: Force user PIN change after set/reset** is set to **1** (the default setting), the Crypto Officer must change the initial CO credential before using the partition for cryptographic operations. This applies to the activation challenge secret as well (see "role changepw" on page 1 in the *LunaCM Command Reference Guide*).

To set up an HA group

1. Create a new HA group, specifying the following information (see "hagroup creategroup" on page 1):
   - the group label (do not call the group "HA")
   - the Serial number OR the slot number of the first member partition
   - the Crypto Officer password or challenge secret for the partition

   lunacm:>**hagroup creategroup -label** <label> {**-slot** <slotnum> | **-serialnumber** <serialnum>}

```
lunacm:> hagroup creategroup -label myHAgroup -slot 0

        Enter the password: ********

        New group with label "myHAgroup" created with group number 1154438865287.
        Group configuration is:

         HA Group Label:  myHAgroup
        HA Group Number:  1154438865287
       HA Group Slot ID:  Not Available
        Synchronization:  enabled
          Group Members:  154438865287
              Needs sync:  no
        Standby Members:  <none>


Slot #    Member S/N                   Member Label    Status
======    ==========                   ============    ======
     0  154438865287                           par0     alive


Command Result : No Error
```

   LunaCM generates a serial number for the HA group (by adding a "1" before the partition serial number), assigns it a virtual slot number, and automatically restarts.

```
lunacm (64-bit) v7.3.0-165. Copyright (c) 2018 SafeNet. All rights reserved.


        Available HSMs:

        Slot Id ->              0
        Label ->                par0
        Serial Number ->        154438865287
        Model ->                LunaSA 7.3.0
        Firmware Version ->     7.3.0
        Configuration ->        Luna User Partition With SO (PW) Key
                                Export With Cloning Mode
```

```
        Slot Description ->     Net Token Slot

        Slot Id ->              1
        Label ->                par1
        Serial Number ->        1238700701509
        Model ->                LunaSA 7.3.0
        Firmware Version ->     7.3.0
        Configuration ->        Luna User Partition With SO (PW) Key
                                Export With Cloning Mode
        Slot Description ->     Net Token Slot

        Slot Id ->              5
        HSM Label ->            myHAgroup
        HSM Serial Number ->    1154438865287
        HSM Model ->            LunaVirtual
        HSM Firmware Version -> 7.3.0
        HSM Configuration ->    Luna Virtual HSM (PW) Key Export With
                                Cloning Mode
        HSM Status ->           N/A - HA Group


Current Slot Id: 0
```

2. Add another partition to the HA group, specifying either the slot or the serial number ("hagroup addmember" on page 1) If the new member contains cryptographic objects, you are prompted to decide whether to replicate the objects within the HA group, or delete them.

   lunacm:>**hagroup addmember -group** <grouplabel> {**-slot** <slotnum> | **-serialnumber** <serialnum>}

```
lunacm:> hagroup addmember -group myHAgroup -slot 1


        Enter the password: ********


Warning:  There are objects currently on the new member.
          Do you wish to propagate these objects within the HA
          group, or remove them?

          Type 'copy' to keep and propagate the existing
          objects, 'remove' to remove them before continuing,
          or 'quit' to stop adding this new group member.
          >  copy

        Member 1238700701509 successfully added to group myHAgroup. New group
        configuration is:

         HA Group Label:  myHAgroup
        HA Group Number:  1154438865287
       HA Group Slot ID:  5
        Synchronization:  enabled
          Group Members:  154438865287, 1238700701509
             Needs sync:  no
        Standby Members:  <none>


Slot #    Member S/N                    Member Label    Status
======    ==========                    ============    ======
     0  154438865287                           par0     alive
     1  1238700701509                          par1     alive
```

```
        Please use the command "ha synchronize" when you are ready
        to replicate data between all members of the HA group.
        (If you have additional members to add, you may wish to wait
        until you have added them before synchronizing to save time by
        avoiding multiple synchronizations.)

Command Result : No Error
```

Repeat this step for each additional HA group member.

3. If you are adding member partitions that already have cryptographic objects stored on them, initiate a manual synchronization. You can tell whether this step is required by checking the line **Needs Sync : yes/no** in the HA group output. This will also confirm that the HA group is functioning correctly ("hagroup synchronize" on page 1).

   lunacm:>**hagroup synchronize -group** <grouplabel>

4. [Optional] If you created an HA group out of empty partitions, and you want to verify that the group is functioning correctly, see "Verifying an HA Group" below.
5. Specify which member partitions, if any, will serve as standby members.

   See "Setting an HA Group Member to Standby" on page 142.

6. Set up and configure auto-recovery (recommended). If you choose to use manual recovery, you will have to execute a recovery command whenever a group member fails.

   See "Configuring HA Auto-Recovery" on page 144.

7. [Optional] Enable HA Only mode (recommended).

   See "Enabling/Disabling HA Only Mode" on page 145.

8. [Optional] Configure HA logging.

   See "HA Logging" on page 146 for procedures and information on reading HA logs.

The HA group is now ready for your application.

Verifying an HA Group

After creating an HA group in LunaCM, you can see the group represented as a virtual slot alongside the physical slots:

```
lunacm (64-bit) v7.3.0-165. Copyright (c) 2018 SafeNet. All rights reserved.


        Available HSMs:

        Slot Id ->              0
        Label ->                par0
        Serial Number ->        154438865287
        Model ->                LunaSA 7.3.0
        Firmware Version ->     7.3.0
        Configuration ->        Luna User Partition With SO (PW) Key Export With Cloning Mode
        Slot Description ->     Net Token Slot

        Slot Id ->              1
        Label ->                par1
        Serial Number ->        1238700701509
        Model ->                LunaSA 7.3.0
        Firmware Version ->     7.3.0
        Configuration ->        Luna User Partition With SO (PW) Key Export With Cloning Mode
```

```
        Slot Description ->     Net Token Slot

        Slot Id ->              5
        HSM Label ->            myHAgroup
        HSM Serial Number ->    1154438865287
        HSM Model ->            LunaVirtual
        HSM Firmware Version -> 7.3.0
        HSM Configuration ->    Luna Virtual HSM (PW) Key Export With Cloning Mode
        HSM Status ->           N/A - HA Group


Current Slot Id: 0
```

The following procedure is one way to verify that your HA group is working as intended:

To verify an HA group

1. Exit LunaCM and run **multitoken** against the HA group slot number (slot 5 in the example) to create some objects on the HA group partitions.

   ./multitoken -mode <keygen_mode> **-key** <key_size> **-nodestroy -slots** <HA_virtual_slot>

   ```
   c:\Program Files\SafeNet\LunaClient>multitoken -mode rsakeygen -key 4096 -nodestroy -slots 5
   multitoken (64-bit) v7.3.0-165. Copyright (c) 2018 SafeNet. All rights reserved.

   Initializing library...Finished Initializing
   ...done.


   Do you wish to continue?

   Enter 'y' or 'n': y


   Constructing thread objects.
   Logging in to tokens...
     slot 0...  Enter password: userpin
       Serial Number 154438865287


   Please wait, creating test threads.

   Test threads created successfully. Press ENTER to terminate testing.

        RSA key generation  4096-bit:

        Using token objects.

           +       keys/second | elapsed
    0,  0 |   total   average  | time (secs)
   ------ | ------- ---------- | ------------
      0.6 |     0.6     0.599* |            5


   Waiting for threads to terminate.
   ```

   You can hit **Enter** at any time to stop the process before the partitions fill up completely. Any number of created objects will be sufficient to show that the HA group is functioning.

2. Run LunaCM and check the partition information on the two physical slots. Check the object count under "Partition Storage":

lunacm:>**partition showinfo**

```
        Current Slot Id: 0

lunacm:> partition showinfo

...(clip)...

        Partition Storage:
                Total Storage Space:  325896
                Used Storage Space:   22120
                Free Storage Space:   303776
                Object Count:         14
                Overhead:             9648

Command Result : No Error


lunacm:> slot set slot 1

        Current Slot Id:    1     (Luna User Slot 7.0.1 (PW) Signing With Cloning Mode)

Command Result : No Error


lunacm:> partition showinfo

...(clip)...

        Partition Storage:
                Total Storage Space:  325896
                Used Storage Space:   22120
                Free Storage Space:   303776
                Object Count:         14
                Overhead:             9648

Command Result : No Error
```

3. To remove the test objects, login to the HA virtual slot and clear the virtual partition ("slot set" on page 1, "partition login" on page 1, "partition clear" on page 1).

   lunacm:>**slot set slot** <HA_virtual_slot>

   lunacm:>**partition login**

   lunacm:>**partition clear**

```
lunacm:> slot set slot 5

        Current Slot Id:    5     (Virtual HSM 7.3.0 (PW) Key Export With Cloning Mode)

Command Result : No Error


lunacm:> partition login

        Option -password was not supplied.  It is required.

        Enter the password: ********

Command Result : No Error


lunacm:> partition clear
```

```
        You are about to delete all the user objects.
        Are you sure you wish to continue?

        Type 'proceed' to continue, or 'quit' to quit now -> proceed

        14 objects were deleted.

Command Result : No Error
```

If you are satisfied that your HA group is working, you can begin using your application against the HA virtual slot. The virtual slot assignment will change depending on how many more application partitions are added to your client configuration. If your application invokes the HA group label, this will not matter. If you have applications that invoke the slot number, see "Enabling/Disabling HA Only Mode" on page 145.

Setting an HA Group Member to Standby

Some HA group members can be designated as standby members. Standby members do not perform any cryptographic operations unless all active members have failed (see "Standby Members" on page 131 for details). They are useful as a last resort against loss of application service.

Prerequisites

The partition you want to designate as a standby member must already be a member of the HA group (see "Adding/Removing an HA Group Member" on page 150). The Crypto Officer must perform this procedure.

To set an HA group member to standby

1. [Optional] Check the serial number of the member you wish to set to standby mode (see "hagroup listgroups" on page 1).

   lunacm:> **hagroup listgroups**

```
lunacm:> hagroup listgroups

        If you would like to see synchronization data for group myHAgroup,
        please enter the password for the group members. Sync info
        not available in HA Only mode.

        Enter the password: ********


             HA auto recovery:  disabled
             HA recovery mode:  activeBasic
Maximum auto recovery retry:  0
Auto recovery poll interval:  60 seconds
                HA logging:  disabled
          Only Show HA Slots:  no

              HA Group Label:  myHAgroup
             HA Group Number:  11238700701509
            HA Group Slot ID:  5
             Synchronization:  enabled
               Group Members:  154438865287, 1238700701509
                  Needs sync:  no
             Standby Members:  <none>


Slot #    Member S/N                    Member Label    Status
```

```
======    ==========                    ============    ======
     0  154438865287                           par0     alive
     1  1238700701509                          par1     alive
     2  2855496365544                          par2     alive


Command Result : No Error
```

2. Set the desired member to standby mode by specifying the serial number (see "hagroup addstandby" on page 1).

   lunacm:> **hagroup addstandby -group** <label> **-serialnumber** <member_serialnum>

```
lunacm:> hagroup addstandby -group myHAgroup -serialnumber 2855496365544

        The member 2855496365544 was successfully added to the standby list for the HA Group
myHAgroup.


Command Result : No Error
```

To remove an HA group member from standby

1. [Optional] Check the serial number of the standby member (see "hagroup listgroups" on page 1).

   lunacm:> **hagroup listgroups**

```
lunacm:> hagroup listgroups

        If you would like to see synchronization data for group myHAgroup,
        please enter the password for the group members. Sync info
        not available in HA Only mode.

        Enter the password: ********


             HA auto recovery:  disabled
             HA recovery mode:  activeBasic
   Maximum auto recovery retry:  0
   Auto recovery poll interval:  60 seconds
                    HA logging:  disabled
            Only Show HA Slots:  no

               HA Group Label:  myHAgroup
              HA Group Number:  11238700701509
             HA Group Slot ID:  5
              Synchronization:  enabled
                Group Members:  154438865287, 1238700701509
                   Needs sync:  no
             Standby Members:  2855496365544


Slot #    Member S/N                    Member Label    Status
======    ==========                    ============    ======
     0  154438865287                           par0     alive
     1  1238700701509                          par1     alive
     2  2855496365544                          par2     alive
```

2. Remove the member from standby and return it to active HA use (see "hagroup removestandby" on page 1).

   lunacm:> **hagroup removestandby -group** <label> **-serialnumber** <member_serialnum>

```
lunacm:> hagroup removestandby -group myHAgroup -serialnumber 2855496365544

        The member 2855496365544 was successfully removed from the standby list for the HA
Group myHAgroup.


Command Result : No Error
```

Configuring HA Auto-Recovery

When auto-recovery is enabled, SafeNet Luna HSM Client performs periodic recovery attempts when it detects a member failure. HA auto-recovery is disabled by default for new HA groups. To enable it, you must set a maximum number of recovery attempts. You can also set the frequency of recovery attempts, and the auto-recovery mode (**activeBasic** or **activeEnhanced**). These settings will apply to all HA groups configured on the client.

To configure HA auto-recovery

1. Set the desired number of recovery attempts by specifying the retry count as follows ("hagroup retry" on page 1):
   - Set a value of **0** to disable HA auto-recovery
   - Set a value of **-1** for unlimited retries
   - Set any specific number of retries from **1** to **500**

   lunacm:> **hagroup retry -count** <retries>

   ```
   lunacm:> hagroup retry -count -1

           HA Auto Recovery Count has been set to -1

   Command Result : No Error
   ```

2. [Optional] Set the desired frequency of recovery attempts by specifying the time in seconds ("hagroup interval" on page 1). The acceptable range is 60-1200 seconds (default: 60).

   lunacm:> **hagroup interval -interval** <seconds>

   ```
   lunacm:> hagroup interval -interval 120

           HA Auto Recovery Interval has been set to 120 seconds.

   Command Result : No Error
   ```

3. [Optional] Set the auto-recovery mode ("hagroup recoverymode" on page 1). The default is **activeBasic**.

   lunacm:> **hagroup recoverymode -mode** {**activeBasic** | **activeEnhanced**}

   ```
   lunacm:> hagroup recoverymode -mode activeEnhanced

           HA Auto Recovery Mode has been set to activeEnhanced mode.

   Command Result : No Error
   ```

4. [Optional] Check that auto-recovery has been enabled ("hagroup listgroups" on page 1). You are prompted for the Crypto Officer password/challenge secret.

   lunacm:> **hagroup listgroups**

   ```
   lunacm:> hagroup listgroups
   ```

```
        If you would like to see synchronization data for group myHAgroup,
        please enter the password for the group members. Sync info
        not available in HA Only mode.

        Enter the password: ********

              HA auto recovery:  enabled
              HA recovery mode:  activeEnhanced
  Maximum auto recovery retry:  infinite
  Auto recovery poll interval:  120 seconds
                    HA logging:  disabled
            Only Show HA Slots:  no
```

Enabling/Disabling HA Only Mode

By default, the client lists both physical slots and virtual HA slots. Directing applications at the physical slots bypasses the high availability and load balancing functionality. An application must be directed at the virtual HA slot to use HA load balancing and redundancy. HA Only mode hides the physical slots and leaves only the HA group slots visible to applications, simplifying the PKCS#11 slot numbering (see "Slot Numbering and Behavior" on page 266).

If an HA group member partition fails and is recovered, all visible slot numbers can change, including the HA group virtual slots. This can cause applications to direct operations to the wrong slot. If a physical slot in the HA group receives a direct request, the results will not be replicated on the other partitions in the group (see "HA Troubleshooting" on page 158) When HA Only mode is enabled, the HA virtual slots are not affected by partition slot changes. Thales Group recommends enabling HA Only mode on all clients running HA groups.

> **NOTE** Individual partition slots are still visible in LunaCM when HA Only mode is enabled. They are hidden only from client applications. Use **CKdemo** (Option **11**) to see the slot numbers to use with client applications.

To enable HA Only mode

1. Enable HA Only mode in LunaCM ("hagroup haonly" on page 1).

   lunacm:> **hagroup haonly -enable**

   ```
   lunacm:> hagroup haonly -enable

           "HA Only" has been enabled.

   Command Result : No Error
   ```

2. [Optional] Since LunaCM still displays the partitions, you can check the status of HA Only mode at any time ("hagroup haonly" on page 1).

   lunacm:> **hagroup haonly -show**

   ```
   lunacm:> hagroup haonly -show

           This system is configured to show only HA slots.  (HA Only is enabled)


   Command Result : No Error
   ```

To disable HA Only mode

1. Disable HA Only mode in LunaCM ("hagroup haonly" on page 1).

   lunacm:> **hagroup haonly -disable**

```
lunacm:> hagroup haonly -disable

        "HA Only" has been disabled.

Command Result : No Error
```

HA Logging

Logging of HA-related events takes place on the Luna HSM Client workstation. The log file **haErrorLog.txt** shows HA errors, as well as add-member and delete-member events. It does not record status changes of the group as a whole (like adding or removing the group).

The HA log rotates after the configured maximum length is reached. When it finishes writing the current record (even if that record slightly exceeds the configured maximum), the file is renamed to include the timestamp and the next log entry begins a new haErrorLog.txt.

> "Configuring HA Logging" below

> "HA Log Messages" on the next page

Configuring HA Logging

Logging is automatically enabled when you configure an HA group (see "Setting Up an HA Group" on page 136), but you must configure a valid destination path before logging can begin. HA groups are configured on the client using LunaCM. The HA configuration settings are saved to the **Chrystoki.conf** (Linux/Unix) or **crystoki.ini** (Windows) file, as illustrated in the following example:

```
VirtualToken = {
VirtualToken00Label = haGroup1; // The label of the HA group.
VirtualToken00SN = 11234840370164; // The pseudo serial number of the HA group.
VirtualToken00Members = 1234840370164, 1234924189183; // The serial number of the members.
VirtualTokenActiveRecovery = activeEnhanced; // The recovery mode.
}
HASynchronize = {
haGroup1 = 1; // Enable automatic synchronization of objects.
}
HAConfiguration = {
HAOnly = 1; // Enable listing HA groups only via PKCS#11 library.
haLogPath = /tmp/halog; // Base path of the HA log file; i.e., "/tmp/halog/haErrorLog.txt".
haLogStatus = enabled; // Enable HA log.
logLen = 100000000; // Maximum size of HA log file in bytes.
failover_on_deactivation = 1; // if a partition becomes deactivated then the client will
immediately failover and resume its operation on the other HA partitions. This is currently an
alpha feature
reconnAtt = 120; // Number of recovery attempts.
}
HARecovery = {
haGroup1 = 1; // Deprecated in this release as auto recovery will cover the use case. When
cryptoki loads into memory it reads the number and if the number changes (gets incremented) then
cryptoki interprets this as a manual recovery attempt.
}
```

To configure HA logging:

Use the LunaCM command **hagroup halog** (see "hagroup halog" on page 1).

1. Set a valid path for the log directory. You must specify an existing directory.

   lunacm:>**hagroup halog -path** <filepath>

   ```
   lunacm:> hagroup halog -path "C:\Program Files\Safenet\Lunaclient\halog"

           HA Log path successfully set to C:\Program Files\Safenet\Lunaclient\halog.

   Command Result : No Error
   ```

2. [Optional] Set the maximum length for individual log files.

   lunacm:>**hagroup halog -maxlength** <max_file_length>

   ```
   lunacm:> hagroup halog -maxlength 500000

           HA Log maximum file size was successfully set to 500000.

   Command Result : No Error
   ```

3. [Optional] Enable or disable HA logging at any time.

   lunacm:>**hagroup halog -disable**

   lunacm:>**hagroup halog -enable**

   ```
   lunacm:> hagroup halog -disable

           HA Log was successfully disabled.

   Command Result : No Error
   ```

4. [Optional] View the current status of the HA logging configuration.

   lunacm:>**hagroup halog -show**

   ```
   lunacm:> hagroup halog -show

            HA Log: enabled
          Log File: C:\Program Files\Safenet\Lunaclient\halog\haErrorLog.txt
   Max File Length: 500000 bytes
   Command Result : No Error
   ```

HA Log Messages

The following table provides descriptions of the messages generated by the HA sub-system and saved to the HA log. The HA log is saved to the location specified by **haLogPath** in the **Chrystoki.conf** (Linux/Unix) or **crystoki.ini** (Windows) file.

Message Format

Every HA log message has a consistent prefix consisting of the date, time, process id, and serial number (of the affected HA group). For example:

```
Wed Oct  4 16:29:21 2017 : [17469] HA group: 11234840370164 …
```

Message Descriptions

In the message descriptions, the term **connection** refers to the connection between the SafeNet Luna client and the SafeNet Luna Network appliance. A connection is considered **valid** if the appliance responds correctly on the IP address and port. The connection can transition to **invalid** for a number of reasons. Some examples include if the appliance Ethernet cable is detached, if the appliance is shutdown/rebooted, or if the NTLS service is stopped/restarted.

Message ID	Message/Description
HALOG_CONFIGURED_AS_PASSWORD	`<MessagePrefix> configured as a "PASSWORD Based" virtual device` **Description:** Message advising that the virtual partition is password-authenticated. This means that you cannot add a PED-authenticated member to the group.
HALOG_CONFIGURED_AS_PED	`<MessagePrefix> configured as a "PED Based" virtual device` **Description:** Message advising that the virtual partition is PED-authenticated. This means that you cannot add a password-authenticated member to the group.
HALOG_DROPMEMBER	`<MessagePrefix> has dropped member: <SerialNumber>` **Description:** The connection changed from valid to invalid, determined after an HSM command (such as C_Sign) fails.
HALOG_DROPUNRECOVERABLE	`<MessagePrefix> unable to reach member: <SerialNumber>. Manual Recover or Auto Recovery will be able to recover this member` **Description:** The connection is invalid, as determined during a call to C_Initialize.
HALOG_LOGINFAILED	`<MessagePrefix> can not login to member: <SerialNumber>, autorecovery will be disabled. Code: <ErrorCodeHex> : <ErrorCodeString>` **Description:** The connection changed from valid to invalid, as determined during a call to C_Login.
HALOG_MEMBER_DEACTIVATED	`<MessagePrefix> member: <SerialNumber> deactivated` **Description:** The user manually deactivated the partition, as determined after an HSM command (such as C_Sign) fails.
HALOG_MEMBER_NOW_ACTIVATED	`<MessagePrefix> recovery attempt <AttemptNumber> member <SerialNumber> is now activated and will be reintroduce back into the HA group.` **Description:** Additional info about the recovered partition, which was deactivated and is now becoming activated.
HALOG_MEMBER_REVOKED	`<MessagePrefix> member: <SerialNumber> revoked` **Description:** The user manually revoked the partition, as determined during a periodic recovery attempt.
HALOG_MEMBERS_OFFLINE	`<MessagePrefix> all members gone offline.` **Description:** A situation where all members go offline. Recovery is not possible at this point.

Message ID	Message/Description
HALOG_MGMT_ THREAD_START	`<MessagePrefix> management thread started` **Description:** This thread is responsible for managing all members and HA in general while the HA group is active. The thread starts up when the application first launches.
HALOG_MGMT_ THREAD_TERMINATE	`<MessagePrefix> management thread terminated` **Description:** This thread is responsible for managing all members and HA in general while the HA group is active. If the client application shuts down, this thread will simply terminate. The thread will start up again once the application re-launches.
HALOG_NEWMEMBER	`<MessagePrefix> detected new member member: <SerialNumber>` **Description:** The user manually added a member to the HA group without restarting the application, as determined during a periodic recovery attempt.
HALOG_RECOVERED	`<MessagePrefix> recovery attempt <Integer> succeeded for member: <SerialNumber>` **Description:** The connection changed from invalid to valid, as determined during a periodic recovery attempt.
HALOG_RECOVERY_ ATTEMPT_#_ REINTRODUCING	`<MessagePrefix> recovery attempt <AttemptNumber> reintroducing <Number> token objects to recovered token <TokenNumber>` **Description:** Additional info about the recovered partition at which some objects were cloned.

HALOG_ RECOVERYFAILED	`<MessagePrefix> recovery attempt <Integer> failed for member: <SerialNumber>. Code: <ErrorCodeHex> : <ErrorCodeString>.`
	If autorecovery fails, then a second message is logged, as follows:
	`<MessagePrefix> exceeded maximum number of autorecovery attempts for member: <SerialNumber>. Autorecovery will be disabled`
	Description: The connection remains invalid, as determined during a periodic recovery attempt.

HALOG_ RENABLEMEMBER (deprecated)	`<MessagePrefix> Re-enable auto recovery process for member: <SerialNumber>` **Description:** The user manually requested partition recovery, as determined during a periodic recovery attempt before an HSM command.
HALOG_ UNRECOVERABLE (deprecated)	`<MessagePrefix> recovery attempt <Integer> failed for member: <SerialNumber>. Manual Recover or Auto Recovery will not be able to recover this member. Code: <ErrorCodeHex> : <ErrorCodeString>` **Description:** The connection is invalid and is not eligible for recovery.

Message ID	Message/Description
No ID*	`<MessagePrefix> member <SerialNumber> is not activated and is excluded from the HA group` **Description:** The HA member was not activated at the time when a C_Initialize call was made, and is therefore excluded from the HA group. Once the partition is activated, the HA group will attempt an automatic recovery, resulting in one of the two messages below
No ID*	`<MessagePrefix> recovery attempt <SerialNumber> is not activated and cannot be reintroduced back into the HA group\n` **Description:** Recovery failed
No ID*	`<MessagePrefix> recovery attempt <SerialNumber> is now activated and will be reintroduce back into the HA group.\n` **Description:** Recovery succeeded

* You might encounter these extra messages in the HA logs. They were added for HA development testing and therefore have no Message IDs assigned to them. They could duplicate information covered by other log messages as defined above.

Managing Your HA Groups

If you set up your HA groups as recommended, using auto-recovery, they require very little direct maintenance. You can perform the following tasks without pausing your applications:

- You can add or remove a member partition at any time (see "Adding/Removing an HA Group Member" below).
- If you declined to use auto-recovery, you must manually recover group members whenever they fail (see "Manually Recovering a Failed HA Group Member" on page 154).
- If an HSM fails permanently, or is re-initialized, the member partition cannot be recovered (see "Replacing an HA Group Member" on page 155).
- If you want to delete an HA group, see "Deleting an HA Group" on page 158.

For other issues, see "HA Troubleshooting" on page 158.

Adding/Removing an HA Group Member

You can add a new member to an HA group at any time using LunaCM, even if your application is running. Cryptographic objects will be replicated on the new partition and operations will be scheduled according to the load-balancing algorithm (see "Load Balancing" on page 127).

Likewise, you can remove a member at any time, and currently-scheduled operations will fail over to the rest of the group members (see "Failover" on page 129).

NOTE If you remove the partition that was used to create the group, the HA group serial number changes to reflect this. This is to prevent another HA group from being assigned the same serial number as the original. If your application queries the HA group serial number, it must redirect operations to the new serial.

Prerequisites

The new member partition must:

- be assigned to the client and visible in LunaCM
- be initialized with the same domain string/red domain PED key as the other partitions in the group
- have the Crypto Officer role initialized with the same credentials as the other partitions in the group
- be activated and have auto-activation enabled (PED-authenticated)

To add an HA group member

1. Open LunaCM on the client workstation and ensure that the new partition is visible.

```
lunacm (64-bit) v7.3.0-165. Copyright (c) 2018 SafeNet. All rights reserved.


        Available HSMs:

        Slot Id ->              0
        Label ->                par0
        Serial Number ->        154438865287
        Model ->                LunaSA 7.3.0
        Firmware Version ->     7.3.0
        Configuration ->        Luna User Partition With SO (PW) Key Export With Cloning Mode
        Slot Description ->     Net Token Slot

        Slot Id ->              1
        Label ->                par1
        Serial Number ->        1238700701509
        Model ->                LunaSA 7.3.0
        Firmware Version ->     7.3.0
        Configuration ->        Luna User Partition With SO (PW) Key Export With Cloning Mode
        Slot Description ->     Net Token Slot

        Slot Id ->              2
        Label ->                par2
        Serial Number ->        2855496365544
        Model ->                LunaSA 7.3.0
        Firmware Version ->     7.3.0
        Configuration ->        Luna User Partition With SO (PW) Key Export With Cloning Mode
        Slot Description ->     Net Token Slot

        Slot Id ->              5
        HSM Label ->            myHAgroup
        HSM Serial Number ->    1154438865287
        HSM Model ->            LunaVirtual
        HSM Firmware Version -> 7.3.0
        HSM Configuration ->    Luna Virtual HSM (PW) Key Export With Cloning Mode
        HSM Status ->           N/A - HA Group
        HSM Certificates ->     *** Test Certs ***


Current Slot Id: 0
```

2. Add the new partition to the HA group by specifying either the slot or the serial number ("hagroup addmember" on page 1). You are prompted for the Crypto Officer password/challenge secret.

   lunacm:> **hagroup addmember -group** <label> {**-slot** <slotnum> | **-serial** <serialnum>}

```
lunacm:> hagroup addmember -group myHAgroup -slot 2

        Enter the password: ********
        Member 2855496365544 successfully added to group myHAgroup. New group
        configuration is:

         HA Group Label:  myHAgroup
        HA Group Number:  1154438865287
       HA Group Slot ID:  5
       Synchronization: enabled
          Group Members:  154438865287, 1238700701509, 2855496365544
              Needs sync:  no
        Standby Members:  <none>


Slot #    Member S/N                    Member Label    Status
======    ==========                    ============    ======
     0  154438865287                           par0     alive
     1  1238700701509                           par1     alive
     2  2855496365544                           par2     alive


        Please use the command "ha synchronize" when you are ready
        to replicate data between all members of the HA group.
        (If you have additional members to add, you may wish to wait
        until you have added them before synchronizing to save time by
        avoiding multiple synchronizations.)

Command Result : No Error
```

To remove an HA group member

1. Remove the partition from the group by specifying either the slot or the serial number ("hagroup removemember" on page 1).

   lunacm:> **hagroup removemember -group** <label> {**-slot** <slotnum> | **-serial** <serialnum>}

```
lunacm:> hagroup removemember -group myHAgroup -slot 0

        Member 154438865287 successfully removed from group myHAgroup.


        Note: Serial number for the group changed to 11238700701509.
Command Result : No Error
```

> **NOTE** If you remove the partition that was used to create the group, the HA group serial number changes to reflect this. This is to prevent another HA group from being assigned the same serial number as the original. If your application queries the HA group serial number, it must redirect operations to the new serial.

   LunaCM restarts.

```
lunacm (64-bit) v7.3.0-165. Copyright (c) 2018 SafeNet. All rights reserved.
```

```
        Available HSMs:

        Slot Id ->              0
        Label ->                par0
        Serial Number ->        154438865287
        Model ->                LunaSA 7.3.0
        Firmware Version ->     7.3.0
        Configuration ->        Luna User Partition With SO (PW) Key Export With Cloning Mode
        Slot Description ->     Net Token Slot

        Slot Id ->              1
        Label ->                par1
        Serial Number ->        1238700701509
        Model ->                LunaSA 7.3.0
        Firmware Version ->     7.3.0
        Configuration ->        Luna User Partition With SO (PW) Key Export With Cloning Mode
        Slot Description ->     Net Token Slot

        Slot Id ->              2
        Label ->                par2
        Serial Number ->        2855496365544
        Model ->                LunaSA 7.3.0
        Firmware Version ->     7.3.0
        Configuration ->        Luna User Partition With SO (PW) Key Export With Cloning Mode
        Slot Description ->     Net Token Slot

        Slot Id ->              5
        HSM Label ->            myHAgroup
        HSM Serial Number ->    11238700701509
        HSM Model ->            LunaVirtual
        HSM Firmware Version -> 7.3.0
        HSM Configuration ->    Luna Virtual HSM (PW) Key Export With Cloning Mode
        HSM Status ->           N/A - HA Group


Current Slot Id: 0
```

2. [Optional] Check that the partition was removed from the group ("hagroup listgroups" on page 1).

   lunacm:> **hagroup listgroups**

```
lunacm:> hagroup listgroups

        If you would like to see synchronization data for group myHAgroup,
        please enter the password for the group members. Sync info
        not available in HA Only mode.

        Enter the password: ********


             HA auto recovery:  disabled
             HA recovery mode:  activeBasic
   Maximum auto recovery retry:  0
   Auto recovery poll interval:  60 seconds
                    HA logging:  disabled
           Only Show HA Slots:  no

               HA Group Label:  myHAgroup
              HA Group Number:  11238700701509
             HA Group Slot ID:  5
             Synchronization: enabled
               Group Members:  1238700701509, 2855496365544
                  Needs sync:  no
```

```
              Standby Members:  <none>


Slot #    Member S/N                    Member Label    Status
======    ==========                    ============    ======
     1  1238700701509                           par1     alive
     2  2855496365544                           par2     alive


Command Result : No Error
```

Manually Recovering a Failed HA Group Member

Thales Group recommends using auto-recovery for all HA group configurations (see "Configuring HA Auto-Recovery" on page 144). If you do not enable auto-recovery and a member partition fails, or if the recovery retry count expires before the partition comes back online, you must recover the partition manually using LunaCM. You do not need to pause your application(s) to perform a manual recovery; the HA group handles load-balancing and automatically replicates any new or changed keys to the recovered member.

To perform a manual recovery of a failed HA group member

1. [Optional] Ensure that the failed member is available and visible in LunaCM by addressing the problem that caused the failure. Display the HA group to see the failed members ("hagroup listgroups" on page 1). You are prompted for the Crypto Officer password/challenge secret.

   lunacm:>**hagroup listgroups**

```
lunacm:> hagroup listgroups

        If you would like to see synchronization data for group myHAgroup,
        please enter the password for the group members. Sync info
        not available in HA Only mode.


        Enter the password: ********


             HA auto recovery:  disabled
             HA recovery mode:  activeBasic
   Maximum auto recovery retry:  0
   Auto recovery poll interval:  60 seconds
                    HA logging:  disabled
              Only Show HA Slots:  yes


         HA Group Label:  myHAgroup
        HA Group Number:  1154438865287
      HA Group Slot ID:  5
      Synchronization: enabled
         Group Members:  154438865287, 1238700701509
            Needs sync:  no
        Standby Members:  <none>


Slot #    Member S/N                    Member Label    Status
======    ==========                    ============    ======
------  154438865287                            par0     alive
------  1238700701509                    ------------      down
```

2. If you are using a PED-authenticated partition with auto-activation disabled, or if the partition was down for longer than two hours, log in to the partition as Crypto Officer and present the black CO PED key.

   lunacm:>**slot set -slot** <slotnum>

   lunacm:>**role login -name co**

3. Execute the manual recovery command, specifying the HA group label ("hagroup recover" on page 1).

   lunacm:>**hagroup recover**

   ```
   lunacm:> ha recover -g myHAgroup

           Signal sent to HA Group "myHAgroup" to recover.

   Command Result : No Error
   ```

   If you have an application running on the HA group, the failed members will be recovered the next time an operation is scheduled. Load-balancing and key replication is automatic.

4. If you do not currently have an application running, you can manually synchronize the contents of the HA group ("hagroup synchronize" on page 1).

   > **CAUTION!** Never use manual synchronization if you have an application running. The HA group performs this automatically. Using this command on an HA group that is running an application could create conflicting key versions.

   lunacm:>**hagroup synchronize -group** <label>

   ```
   lunacm:> hagroup synchronize -group myHAgroup

           Enter the password: ********

           Synchronization completed.

   Command Result : No Error
   ```

Replacing an HA Group Member

Sometimes an HSM failure is permanent (from the perspective of the HA group). For example, if the HSM is re-initialized, the member partition is erased and must be recreated. In this case, you can recreate a partition on the same HSM or another HSM, and deploy the new member to the group. You do not need to pause your application to replace an HA group member.

Prerequisites

The Crypto Officer must complete this procedure, but any new member partition must first be created and assigned to the client by the HSM SO, and initialized by the Partition SO. All the prerequisites listed in "Setting Up an HA Group" on page 136 must be met.

To replace an HA group member

1. [Optional] Display the HA group to see the failed member ("hagroup listgroups" on page 1). You are prompted for the Crypto Officer password/challenge secret.

   lunacm:>**hagroup listgroups**

   ```
   lunacm:> hagroup listgroups
   ```

```
        If you would like to see synchronization data for group myHAgroup,
        please enter the password for the group members. Sync info
        not available in HA Only mode.


        Enter the password: ********


             HA auto recovery:  enabled
             HA recovery mode:  activeEnhanced
   Maximum auto recovery retry:  500
   Auto recovery poll interval:  60 seconds
                    HA logging:  disabled
            Only Show HA Slots:  yes


         HA Group Label:  myHAgroup
        HA Group Number:  1154438865287
       HA Group Slot ID:  5
       Synchronization: enabled
          Group Members:  154438865287, 1238700701509
             Needs sync:  no
        Standby Members:  <none>


Slot #    Member S/N                    Member Label    Status
======    ==========                    ============    ======
------  154438865287                           par0     alive
------  1238700701509                   ------------      down
```

2. Prepare the new HA group member, whether that means creating a new partition on the original HSM or configuring a new SafeNet Luna PCIe HSM, and assign the new partition to the HA client. Ensure that the new member partition and the HSM on which it resides meet the prerequisites outlined in "Setting Up an HA Group" on page 136 and is visible in LunaCM.

```
lunacm (64-bit) v7.3.0-165. Copyright (c) 2018 SafeNet. All rights reserved.


        Available HSMs:

        Slot Id ->              0
        Label ->                par0
        Serial Number ->        154438865287
        Model ->                LunaSA 7.3.0
        Firmware Version ->     7.3.0
        Configuration ->        Luna User Partition With SO (PW) Key Export With Cloning Mode
        Slot Description ->     Net Token Slot

        Slot Id ->              1
        Label ->                par1
        Serial Number ->        1238700701510
        Model ->                LunaSA 7.3.0
        Firmware Version ->     7.3.0
        Configuration ->        Luna User Partition With SO (PW) Key Export With Cloning Mode
        Slot Description ->     Net Token Slot

        Slot Id ->              5
        HSM Label ->            myHAgroup
        HSM Serial Number ->    1154438865287
        HSM Model ->            LunaVirtual
        HSM Firmware Version -> 7.3.0
```

```
        HSM Configuration ->    Luna Virtual HSM (PW) Key Export With Cloning Mode
        HSM Status ->           N/A - HA Group


Current Slot Id: 0
```

3. Add the new partition to the HA group by specifying either the slot or the serial number ("hagroup addmember" on page 1). You are prompted for the Crypto Officer password/challenge secret.

   lunacm:> **hagroup addmember -group** <label> {**-slot** <slotnum> | **-serial** <serialnum>}

```
lunacm:> hagroup addmember -group myHAgroup -slot 1

        Enter the password: ********
        Member 1238700701510 successfully added to group myHAgroup. New group
        configuration is:

         HA Group Label:  myHAgroup
        HA Group Number:  1154438865287
       HA Group Slot ID:  5
       Synchronization: enabled
          Group Members:  154438865287, 1238700701509, 1238700701510
             Needs sync:  no
        Standby Members:  <none>


Slot #    Member S/N                    Member Label    Status
======    ==========                    ============    ======
     0  154438865287                           par0     alive
------  1238700701509                     ------------      down
     1  1238700701510                           par1     alive


        Please use the command "ha synchronize" when you are ready
        to replicate data between all members of the HA group.
        (If you have additional members to add, you may wish to wait
        until you have added them before synchronizing to save time by
        avoiding multiple synchronizations.)

Command Result : No Error
```

   The new partition is now an active member of the HA group. If you have an application currently running, cryptographic objects are automatically replicated to the new member and it is assigned operations according to the load-balancing algorithm.

4. Remove the old partition from the group by specifying the serial number ("hagroup removemember" on page 1).

   lunacm:> **hagroup removemember -group** <label> **-serial** <serialnum>

```
lunacm:> hagroup removemember -group myHAgroup -serial 1238700701509

        Member 1238700701509 successfully removed from group myHAgroup.

Command Result : No Error
```

   LunaCM restarts.

5. [Optional] If you do not currently have an application running, you can manually synchronize the contents of the HA group ("hagroup synchronize" on page 1).

> **CAUTION!** Never use manual synchronization if you have an application running. The HA group performs this automatically. Using this command on an HA group that is running an application could create conflicting key versions.

lunacm:>**hagroup synchronize -group** <label>

```
lunacm:> hagroup synchronize -group myHAgroup

        Enter the password: ********

        Synchronization completed.

Command Result : No Error
```

6. [Optional] If you intend to have the new partition serve as a standby member, see "Setting an HA Group Member to Standby" on page 142.

Deleting an HA Group

Use LunaCM to delete an HA group from your configuration.

> **NOTE** This procedure only removes the HA group virtual slot; the member partitions and all their contents remain intact. Only the HSM SO can delete individual partitions.

To delete an HA group

1. Stop any applications currently using the HA group.
2. Delete the group by specifying its label (see "hagroup deletegroup" on page 1).

   lunacm:> **hagroup deletegroup -group** <label>

```
lunacm:> hagroup deletegroup -label myHAgroup

        The HA group myHAgroup was successfully deleted.
Command Result : No Error
```

HA Troubleshooting

If you encounter problems with an HA group, refer to this section.

Administration Tasks on HA Groups

Do not attempt to run administrative tasks on an HA group virtual slot (such as changing the CO password or altering partition policies). These virtual slots are intended for cryptographic operations only. It is not possible to use an HA group to make administrative changes to all partitions in the group simultaneously.

Unique Object IDs (OUID)

If two applications using the same HA group modify the same object using different members, the object fingerprint may conflict.

Client-Side Failures

Any failure of the client (such as operating system problems) that does not involve corruption or removal of files, should resolve itself when the client is rebooted.

If the client workstation seems to be working fine otherwise, but you have lost visibility of the HSMs in LunaCM or your client, try the following remedies:

- verify that the Thales Group drivers are running, and retry
- reboot the client workstation
- restore your client configuration from backup
- re-install SafeNet Luna HSM Client and re-configure the HA group

For SafeNet Luna PCIe HSM, the client is the HSM host. If HA has been working, any sudden failure is likely to be OS or driver related (restart) or file corruption (re-install). If a re-install is necessary, you must recreate and reconfigure the HA group.

Effect of PED Operations

PED operations can block some cryptographic operations, so that while a member of an HA group is performing a PED operation, it could appear to the HA group as a failed member. When the PED operation is complete, failover and recovery HA logic are invoked to return the member to normal operation.

CHAPTER 7: HSM Initialization

Initialization prepares a new HSM for use, or an existing HSM for reuse, as follows. You must initialize the HSM before you can generate or store objects, allow clients to connect, or perform cryptographic operations:

- On a new HSM or factory-reset HSM, initialization sets the HSM SO credentials, the HSM label, and the cloning domain of the HSM Admin partition. This is often referred to as a 'hard' initialization. See "Initializing a New or Factory-reset HSM" on the next page.
- On an existing, non-factory-reset HSM, reinitialization destroys all existing partitions and objects, but retains the SO credentials and cloning domain. You have the option to change or retain the existing label. This is often referred to as a 'soft' initialization. See "Re-initializing an Existing, Non-factory-reset HSM" on page 163.

> **NOTE** To ensure accurate auditing, perform initialization only after you have set the system time parameters (time, date, time zone, use of NTP (Network Time Protocol). You can use the **-authtimeconfig** option when initializing the HSM to require HSM SO authorization of any time-related changes once the HSM is initialized.

Hard versus soft initialization

The following table summarizes the differences between a hard and soft initialization.

Condition/Effect	Soft init	Hard init
HSM SO authentication required	Yes	No
Can set new HSM label	Yes	Yes
Creates new HSM SO identity	No	Yes
Creates new Domain	No	Yes
Destroys partitions	Yes	No (none exist to destroy, since the HSM is new or an **hsm factoryreset** was performed)
Destroys objects	Yes	No (none exist to destroy, since the HSM is new or an **hsm factoryreset** was performed)

Initializing a New or Factory-reset HSM

> **NOTE** New HSMs are shipped in Secure Transport Mode (STM). You must recover the HSM from STM before you can initialize the HSM. See "To initialize a new or factory-reset HSM (hard init):" on the next page for details.

On a new, or factory reset HSM (using **hsm factoryreset**), you perform a 'hard init' to set the following:

HSM Label	The label is a string of up to 32 characters that identifies this HSM unit uniquely. A labeling convention that conveys some information relating to business, departmental or network function of the individual HSM is commonly used. Labels cannot contain a leading space.
HSM SO credentials	For PED-authenticated HSMs, you create a new HSM SO (blue) PED key(set) or re-use an existing key(set) from an HSM you want to share credentials with. If you are using PED authentication, ensure that you have a PED key strategy before beginning. See "PED Authentication" on page 187. For password-authenticated HSMs, you specify the HSM SO password. For proper security, it should be different from the appliance admin password, and employ standard password-security characteristics. Password can be between 7 and 256 characters in length: > Valid characters are !#$%'*+,-./0123456789:=?@ABCDEFGHIJKLMNOPQRSTUVWXYZ []^_abcdefghijklmnopqrstuvwxyz{}~ (the first character in that list is the space character) > Invalid characters are "&';<>\`|()
Cloning domain for the HSM Admin partition	The cloning domain is a shared identifier that makes cloning possible among a group of HSM partitions. It specifies the security domain (group of HSM partitions) within which the HSM Admin partition can share cryptographic objects though cloning, backup/restore, or in high availability configurations. Note that the HSM Admin partition cloning domain is independent of the cloning domain specified when creating application partitions on the HSM. For PED-authenticated HSMs, you create a new Domain (red) PED key(set) or re-use an existing key(set) from an HSM you want to be able to clone with. For password-authenticated HSMs, you create a new domain password or re-use an existing password from an HSM you want to be able to clone with. Cloning domain strings can be between 1 and 128 characters in length: > Valid characters are !#$%'*+,-./0123456789:=?@ABCDEFGHIJKLMNOPQRSTUVWXYZ []^_abcdefghijklmnopqrstuvwxyz{}~ (the first character in that list is the space character) > Invalid characters are "&';<>\`|() **NOTE** Always specify a cloning domain when you initialize a Password-authenticated SafeNet Luna HSM in a production environment. The HSM allows you to specify "defaultdomain" at initialization, the factory-default domain. This is deprecated, as it is insecure. Anyone could clone objects to or from such an HSM. The default domain is provided for benefit of customers who have previously used the default domain, and for migration purposes. When you prepare a SafeNet Luna HSM to go into service in a real production environment, always specify a proper, secure domain string when you initialize the HSM.

To initialize a new or factory-reset HSM (hard init):

> **CAUTION!** Ensure that you are prepared. Once initialized, re-initializing the HSM forces the deletion of all partitions and objects on the HSM.

1. If Secure Transport Mode is set, you must unlock the HSM before proceeding. New SafeNet Luna HSMs are shipped from the factory in Secure Transport Mode (STM). STM allows you to verify whether or not an HSM has been tampered while it is not in your possession, such as when it is shipped to another location, or placed into storage. See "Secure Transport Mode" on page 262 in the *Administration Guide* for more information.

   To recover your HSM from Secure Transport Mode, proceed as follows:

   a. As part of the delivery process for your new HSM, you should have received an email from Thales Client Services, containing two 16-digit strings, as follows. You will need both of these strings to recover the HSM from STM:

      Random User String: XXXX-XXXX-XXXX-XXXX

      Verification String: XXXX-XXXX-XXXX-XXXX

   b. Ensure that you have the Random User String and Verification String that were emailed to you for your new HSM.

   c. Enter the following command to recover from STM, specifying the Random User String that was emailed to you for your new HSM:

      lunacm:> **stm recover -randomuserstring** <XXXX-XXXX-XXXX-XXXX>

   d. You are presented with a verification string. If the verification string matches the original verification string emailed to you for your new HSM, the HSM has not been tampered, and can be safely deployed. If the verification string does not match the original verification string emailed to you for your new HSM, the HSM has been tampered while in STM. If the verification strings do not match, contact Thales Group Technical Support immediately.

   e. Enter **proceed** to recover from STM (regardless of whether the strings match or not), or enter **quit** to remain in STM.

2. If you are initializing a PED-authenticated HSM, have the Luna PED connected and ready (via USB, in Local PED-USB mode). If your PED is not in USB mode, see "Changing Modes" on page 195 in the *HSM Administration Guide*.

3. Open a LunaCM session and set the slot to the HSM Admin partition.

4. Run the **hsm init** command, specifying a label for your SafeNet Luna PCIe HSM:

   lunacm:> **hsm init** <label>

5. Respond to the prompts to complete the initialization process:

   - on a password-authenticated HSM, you are prompted for the HSM password and for the HSM Admin partition cloning domain string (cloning domains for application partitions are set when the application partitions are initialized).
   - on a PED-authenticated HSM, you are prompted to attend to the PED to create a new HSM SO (blue) PED key for this HSM, re-use an HSM SO PED key from an existing HSM so that you can also use it to log in to this HSM, or overwrite an existing key with a new PED secret for use with this HSM. You are also

prompted to create, re-use, or overwrite the Domain (red) PED key. You can create MofN quorum keysets and duplicate keys as required. See "PED Authentication" on page 187 for more information.

The prompts are self explanatory. New users (especially those initializing a PED-authenticated HSM) may want to refer to the following examples for more information:

- "PED-authenticated HSM Initialization Example" below
- "Password-authenticated HSM Initialization Example" on page 169

Re-initializing an Existing, Non-factory-reset HSM

On an existing, non-factory-reset HSM, re-initialization clears all existing partitions and objects, but retains the SO credentials and cloning domain. You have the option to change or retain the existing label. Re-initialization is also referred to as a soft init. If you do not want to do a soft init, and also change the SO credentials and cloning domain, you need to use the **hsm factoryreset** command to factory reset the HSM, and then perform the procedure described in "Initializing a New or Factory-reset HSM" on page 161.

> **CAUTION!** Ensure you have backups for any partitions and objects you want to keep, before reinitializing the HSM.

To re-initialize an existing, non-factory-reset HSM (soft init):

1. Log in as the HSM SO.
2. If Secure Transport Mode is set, you must unlock the HSM before proceeding. See "Secure Transport Mode" on page 262 in the *Administration Guide*.
3. If you are initializing a PED-authenticated HSM, have the Luna PED connected and ready (via USB, in Local PED-USB mode). If your PED is not in USB mode, see "Changing Modes" on page 195 in the *HSM Administration Guide*.
4. Open a LunaCM session and set the slot to the HSM Admin partition.
5. Run the **hsm init** command, specifying a label for your SafeNet Luna Network HSM:

   lunacm:> **hsm init** <label>

PED-authenticated HSM Initialization Example

This section provides detailed examples that illustrate your options when initializing a PED-authenticated HSM. It provides the following information:

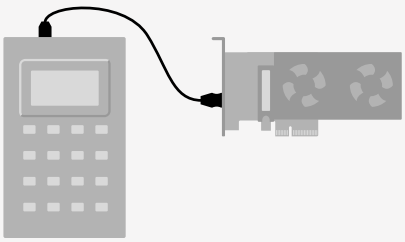
> "To initialize a PED-authenticated HSM:" on the next page
> "Imprinting the Blue HSM SO PED Key" on page 165
> "Imprinting the Red Cloning Domain PED Key" on page 167
> "New, reuse, and overwrite options" on page 168

NOTE Respond promptly to avoid PED timeout Error. If the PED has timed out, press the **CLR** key for five seconds to reset, or switch the PED off, and back on, to get to the “Awaiting command....” state before re-issuing a LunaSH command that invokes the PED.

To initialize a PED-authenticated HSM:

1. Your Luna PED must be connected to the HSM, either locally/directly in USB mode (see "Changing Modes" on page 195), or remotely via Remote PED connection (see "About Remote PED" on page 198).

   NOTE To operate in Local PED-USB mode, the PED must be connected directly to the HSM card's USB port, and not one of the other USB connection ports on the host system.

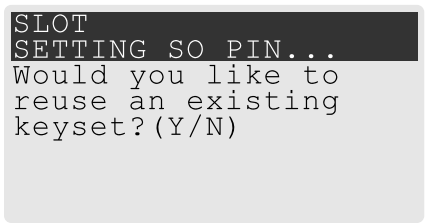
2. Set the active slot to the SafeNet Luna PCIe HSM Admin partition, and issue the **hsm init** command. The HSM passes control to the Luna PED, and the command line directs you to attend to the PED prompts.
3. When you issue the **hsm init** command, the HSM passes control to the Luna PED, and the command line (lunash:>) directs you to attend to the PED prompts.
4. A "default" login is performed, just to get started (you don't need to supply any authentication for this step).
5. Luna PED asks: "Do you wish to reuse an existing keyset?". If the answer is **No**, the HSM creates a new secret which will reside on both the HSM and the key (or keys) that is (or are) about to be imprinted. If the answer is **Yes**, then the HSM does not create a new secret and instead waits for one to be presented via the PED.
6. Luna PED requests a blue PED key. It could be blank to begin with, or it could have a valid secret from another HSM (a secret that you wish to preserve), or it could have a secret that is no longer useful.
7. Luna PED checks the key you provide. If the PED key is not blank, and your answer to "...reuse an existing keyset" was **Yes**, then Luna PED proceeds to copy the secret from the PED key to the HSM.
8. If the key is not blank, and your answer to "...reuse an existing keyset" was **No**, then the PED inquires if you wish to overwrite its contents with a new HSM secret. If the current content of the key is of no value, you say **Yes**. If the current content of the key is a valid secret from another HSM (or if you did not expect the key to hold any data) you can remove it from the PED and replace it with a blank key or a key containing non-useful data, before you answer **Yes** to the 'overwrite' question.
9. Assuming that you are using a new secret, and not reusing an existing one, Luna PED asks if you wish to split the new HSM secret. It does this by asking for values of "M" and "N". You set those values to "1" and "1" respectively, unless you require MofN split-secret, multi-person quorum access control for your HSM (See "M of N Split Secrets (Quorum)" on page 192 for details).
10. Luna PED asks if you wish to use a PED PIN (an additional secret; see "PED Key Management" on page 211 for more info).

11. If you just press **Enter** (effectively saying 'no' to the PED PIN option), then the secret generated by the HSM is imprinted on the PED key, that same secret is retained as-is on the HSM, and the same secret becomes the piece needed to unlock the Security Officer/HSM Admin account on the HSM.
12. If you press some digits on the PED keypad (saying 'yes' to the PED PIN option), then the PED combines the HSM-generated secret with your PED PIN and feeds the combined data blob to the HSM. The HSM throws away the original secret and takes on the new, combined secret as its SO/HSM Admin secret.
13. The PED key contains the original HSM-generated secret, but also contains the flag that tells the PED whether to demand a PED PIN (which is either no digits, or a set of digits that you supplied, and must supply at all future uses of that PED key).
14. Luna PED gives you the option to create some duplicates of this imprinted key. You should make at least one duplicate for backup purposes. Make additional duplicates if your security policy permits, and your procedures require them.
15. Next, Luna PED requests a red Domain PED key. The HSM provides a cloning Domain secret and the PED gives you the option to imprint the secret from the HSM, or to use a domain that might already be on the key. You choose appropriately. If you are imprinting a new Domain secret, you have the same opportunities to split the secret, and to apply a PED PIN "modifier" to the secret. Again, you are given the option to create duplicates of the key.
16. At this point, the HSM is initialized and Luna PED passes control back to LunaCM.

Further actions are needed to prepare for use by your Clients, but you can now log in as SO/HSM Admin and perform HSM administrative actions.

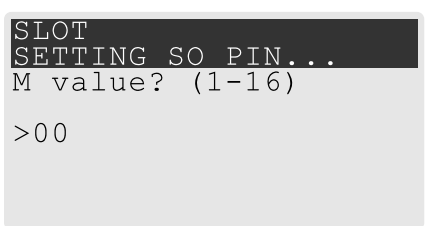
Imprinting the Blue HSM SO PED Key

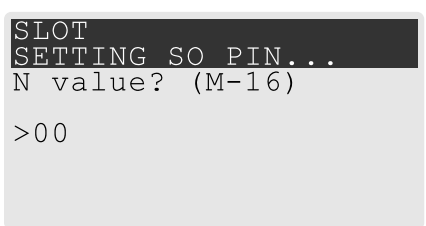
1. Decide if you want to reuse a keyset.



- If you say **No** (on the PED keypad), then you are indicating there is nothing of value on your PED keys to preserve, or you are using blank keys.
- If you say **Yes**, you indicate that you have a PED key (or set of PED keys) from another HSM and you wish your current/new HSM to share the authentication with that other HSM. Authentication will be read from the PED key that you present and imprinted onto the current HSM.

2. Set MofN.

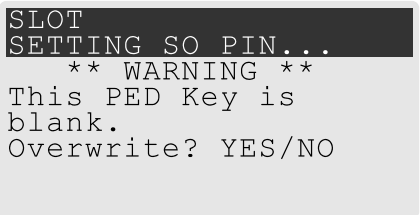




- Setting M and N to **1** means that the role authentication is not to be split, and only a single PED key will be necessary when the authentication is called for in future. Input **1** for each prompt if you do not want to use MofN.
- Setting M and N to larger than 1 sets a quorum requirement for the role, which means that the authentication is split into N different splits, of which quantity M of them (the quorum) must be presented each time you are required to authenticate. MofN allows you to enforce multi-person access control - no single person can access the HSM without cooperation of a quorum of other holders.

3. Insert your blank key or the key you wish to overwrite.

Insert a blue HSM Admin/SO PED key and press **Enter**.

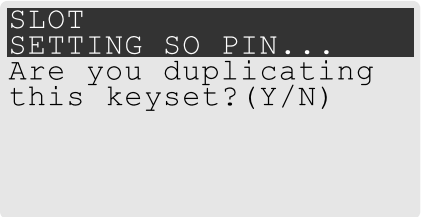


- **Yes**: If the PED should overwrite the PED key with a new SO authentication.
  If you overwrite a PED key that contains authentication secret for another HSM, then this PED key will no longer be able to access the other HSM, only the new HSM that you are currently initializing with a new, unique authentication secret .
- **No**: If you have changed your mind or inserted the wrong PED key.

4. For any situation other than reusing a keyset, Luna PED now prompts for you to set a PED PIN. For multi-factor authentication security, the physical PED key is "something you have." You can choose to associate that with "something you know," in the form of a multi-digit PIN code that must always be supplied along with the PED key for all future HSM access attempts.

Type a numeric password on the PED keypad, if you wish. Otherwise, just press **Enter** twice to indicate that no PED PIN is desired.

5. Decide if you want to duplicate your keyset.



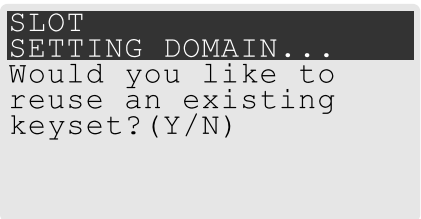
- **Yes**: Present one or more blank keys, all of which will be imprinted with exact copies of the current PED key's authentication.
- **No**: Do not make any copies.

> **NOTE** You should always have backups of your imprinted PED keys, to guard against loss or damage.

Imprinting the Red Cloning Domain PED Key

To begin imprinting a Cloning Domain (red PED key), you must first log into the HSM. Insert your blue SO PED key.

1. Decide if you want to reuse a keyset.



   - **No**: If this is your first SafeNet Luna HSM, or if this HSM will not be cloning objects with other HSMs that are already initialized
   - **Yes**: If you have another HSM and wish that HSM and the current HSM to share their cloning Domain.
2. Set MofN.
   - Setting M and N to **1** means that the domain authentication is not to be split, and only a single PED key will be necessary when the authentication is called for in future. Input **1** for each prompt if you do not want to use MofN.
   - Setting M and N to larger than 1 sets a quorum requirement for the domain, which means that the authentication is split into N different splits, of which quantity M of them (the quorum) must be presented each time you are required to provide the domain. MofN allows you to enforce multi-person access control - no single person can access the HSM without cooperation of a quorum of other holders.
3. Insert your blank key or the key you wish to overwrite.
4. Optionally set a PED PIN.
5. Decide if you want to duplicate your keyset.

Once you stop duplicating the Domain key, or you indicate that you do not wish to make any duplicates, Luna PED goes back to "Awaiting command...". LunaSH says:

```
Command Result : No Error
```

New, reuse, and overwrite options

The table below summarizes the steps involving Luna PED immediately after you invoke the command **hsm init**. The steps in the table are in the order in which they appear as PED prompts, descending down the column.

The first column is the simplest, and most like what you would encounter the very first time you initialize, using "fresh from the carton" PED keys.

The next two columns of the table show some differences if you are using previously-imprinted PED keys, choosing either to reuse what is found on the key (imprint it on your new HSM - see "Shared PED Key Secrets" on page 191) or, to overwrite what is found and generate a new secret to be imprinted on both the PED key and the HSM.

New PED Keys	Existing PED Keys (Reuse)	Existing PED Keys (Overwrite)
SLOT 01 SETTING SO PIN... Would you like to reuse an existing keyset? (Y/N) **No**	SLOT 01 SETTING SO PIN... Would you like to reuse an existing keyset? (Y/N) **Yes**	SLOT 01 SETTING SO PIN... Would you like to reuse an existing keyset? (Y/N) **No**
SLOT 01 SETTING SO PIN... Insert a SO / HSM Admin PED Key Press ENTER.	SLOT 01 SETTING SO PIN... Insert a SO / HSM Admin PED Key Press ENTER.	Slot 01 SETTING SO PIN... Insert a SO / HSM Admin PED Key Press ENTER.
This PED Key is blank. Overwrite? (YES/NO) **Yes**	****Warning!**** This PED Key is for SO / HSM Admin Overwrite? (YES/NO) **No**	****Warning!**** This PED Key is for SO / HSM Admin Overwrite? (YES/NO) **Yes**
Enter a new PED PIN Confirm new PED PIN > Press **Enter** for no PED PIN > Input 4-16 digits on the PED keypad	Enter a new PED PIN Confirm new PED PIN > Press **Enter** for no PED PIN > Input 4-16 digits on the PED keypad	Enter a new PED PIN Confirm new PED PIN > Press **Enter** for no PED PIN > Input 4-16 digits on the PED keypad
Are you duplicating this keyset? YES/NO > **Yes**: duplicate. This option can be looped for as many duplicates as you need > **No**: do not duplicate	Are you duplicating this keyset? YES/NO > **Yes**: duplicate. This option can be looped for as many duplicates as you need > **No**: do not duplicate	Are you duplicating this keyset? YES/NO > **Yes**: duplicate. This option can be looped for as many duplicates as you need > **No**: do not duplicate

New PED Keys	Existing PED Keys (Reuse)	Existing PED Keys (Overwrite)
Login SO / HSM Admin... Insert a SO/ HSM Admin PED Key Press ENTER	Login SO / HSM Admin.. Insert a SO/ HSM Admin PED Key Press ENTER	Login SO / HSM Admin.. Insert a SO/ HSM Admin PED Key Press ENTER
SETTING DOMAIN... Would you like to reuse an existing keyset? (Y/N) > **Yes** (unless you have good reason to create a new domain)	SETTING DOMAIN... Would you like to reuse an existing keyset? (Y/N) > **Yes**: make this HSM part of an existing domain > **No**: create a new domain for this HSM	SETTING DOMAIN... Would you like to reuse an existing keyset? (Y/N) > **Yes**: make this HSM part of an existing domain > **No**: create a new domain for this HSM

Password-authenticated HSM Initialization Example

```
lunacm:>hsm init -label myLunaHSM

        You are about to initialize the HSM.
        All contents of the HSM will be destroyed.

        Are you sure you wish to continue?

        Type 'proceed' to continue, or 'quit' to quit now ->proceed

        Enter password for SO: ********

        Re-enter password for SO: ********

        Option -domain was not specified.  It is required.

        Enter the domain name: *********

        Re-enter the domain name: *********

Command Result : No Error
```

When activity is complete, the system displays a “success” message.

CHAPTER 8: HSM Status Values

Each HSM administrative slot shown in a LunaCM slot listing includes an HSM status. Here are the possible values and what they mean, and what is required to recover from each one.

Indicated Status of HSM	Meaning	Recovery
OK	The HSM is in a good state, working properly.	n/a
Zeroized	The HSM is in zeroized state. All objects and roles are unusable.	HSM initialization is required before the HSM can be used again. "Hard init" - HSM SO and domain are gone, no authentication required. (see Note1)
Decommissioned	The HSM has been decommissioned.	HSM initialization is required before the HSM can be used again. "Hard init" - HSM SO and domain are gone, no authentication required. (see Note1)
Transport Mode	The HSM is in Secure Transport Mode.	STM must be disabled before the HSM can be used.
Transport Mode, zeroized	The HSM is in Secure Transport Mode, and is also zeroized.	STM must be disabled, and then HSM initialization is required before the HSM can be used.
Transport Mode, Decommissioned	The HSM is in Secure Transport Mode, and has been decommissioned.	STM must be disabled, and then HSM initialization is required before the HSM can be used.
Hardware Tamper	The HSM has been tampered. (MTK is destroyed and must be rebuilt from recovery vectors.)	Reboot the host or restart the HSM (vreset for SafeNet Luna PCIe HSM, or ureset for SafeNet Luna USB HSM). The event is logged
Hardware Tamper, Zeroized	The HSM has been tampered. (MTK is destroyed and must be rebuilt from recovery vectors.) The HSM is also in zeroized state. All objects and roles are unusable.	Reboot the host or restart the HSM (vreset for SafeNet Luna PCIe HSM, or ureset for SafeNet Luna USB HSM). The event is logged. HSM initialization is required before the HSM can be used again. HSM SO and domain are gone, no authentication required. (see Note1)

Indicated Status of HSM	Meaning	Recovery
HSM Tamper, Decommissioned	The HSM has been tampered. (MTK is destroyed and must be rebuilt from recovery vectors.) The HSM has also been decommissioned.	Reboot the host or restart the HSM (vreset for SafeNet Luna PCIe HSM, or ureset for SafeNet Luna USB HSM). The event is logged. HSM initialization is required before the HSM can be used again. HSM SO and domain are gone, no authentication required. (see Note1)

NOTE1: A condition, not reported above, preserves the HSM SO and the associated Domain, while SO objects and all application partitions and contents are destroyed. In this case, HSM SO login is required to perform a "soft init". See "HSM Initialization" on page 160 for more information.

For a comparison of various destruction or denial actions on the HSM, see "Comparison of Destruction/Denial Actions" on page 121.

CHAPTER 9: Keys In Hardware vs. Private Key Export

By default, the SafeNet Luna PCIe HSM stores all keys in hardware, allowing private keys to be copied only to another SafeNet Luna HSM (cloning). Cloning allows you to move or copy key material from the HSM to a backup HSM or to another HSM in the same HA group. You might, however, want to export private keys to an encrypted file for off-board storage or use. Individual partitions can be configured in one of three modes for handling private keys.

> **NOTE** This feature has software and/or firmware dependencies. See "Version Dependencies by Feature" on page 317 for more information.

The Partition SO can set the mode by changing the following policies (see "Partition Capabilities and Policies" on page 87 for more information):

- **Partition policy 0: Allow private key cloning** (default: **1**)
- **Partition policy 1: Allow private key wrapping** (default: **0**)

> **NOTE** These partition policies can never be set to **1** (ON) at the same time. An error will result (CKR_CONFIG_FAILS_DEPENDENCIES).

The policies can be set at the time of initialization, using a policy template (see "Policy Templates" on page 93) or by following the procedures described below:

- "Cloning Mode" below
- "Key Export Mode" on the next page
- "No Backup Mode" on page 174

> **NOTE** Partition configurations are listed in LunaCM as "Key Export With Cloning Mode". This indicates that the partition is capable of being configured for either Key Export or Cloning, with the mode of operation defined by the policies listed above. You can never configure a partition to allow both export and cloning of private keys at once.

Cloning Mode

A partition in Cloning mode has the following capabilities and restrictions:

- All keys/objects can be cloned to another partition or SafeNet Luna Backup HSM in the same cloning domain.
- All keys/objects are replicated within the partition's HA group.
- Private keys cannot be wrapped off the HSM (cannot be exported to a file encrypted with a wrapping key).

In this mode, private keys are never allowed to exist outside of a trusted SafeNet Luna HSM in the designated cloning domain. Cloning mode is the default setting for new partitions.

Setting Cloning Mode on a Partition

Cloning mode is the default setting on new partitions. If another mode was set previously, the Partition SO can use the following procedure to set Cloning mode. Use **partition showpolicies** to see the current policy settings.

> **CAUTION!** **Partition policy 0: Allow private key cloning** is Off-to-On destructive by default. Back up any important cryptographic material on the partition before continuing. This destructiveness setting can be customized by initializing the partition with a policy template (see "Editing a Policy Template" on page 94).

To manually set Cloning mode on a partition:

1. Log in to the partition as Partition SO.

   lunacm:>**slot set slot** <slotnum>

   lunacm:>**role login -name po**

2. Set **partition policy 1: Allow private key wrapping** to **0** (OFF).

   lunacm:>**partition changepolicy -policy 1 -value 0**

3. Set **partition policy 0: Allow private key cloning** to **1** (ON).

   lunacm:>**partition changepolicy -policy 0 -value 1**

To initialize a partition in Cloning mode using a policy template:

Use a standard text editor to include the following lines in the policy template file (see "Editing a Policy Template" on page 94):

```
0:"Allow private key cloning":1:1:0
1:"Allow private key wrapping":0:1:0
```

Key Export Mode

A partition in Key Export mode has the following capabilities and restrictions:

- Private keys cannot be cloned to other partitions nor to a SafeNet Luna Backup HSM.
- The partition cannot be part of an HA group (private keys will not be replicated).
- All keys/objects, including private keys, can be wrapped off the HSM (can be exported to a file encrypted with a wrapping key).

This mode is useful when generating key pairs for identity issuance, where transient key-pairs are generated, wrapped off, and embedded on a device. They are not used on the HSM, but generated and issued securely, and then deleted from the HSM.

Setting Key Export Mode on a Partition

The Partition SO can use the following procedure to set Key Export mode. Use **partition showpolicies** to see the current policy settings.

> **CAUTION!** **Partition policy 1: Allow private key wrapping** is always Off-to-On destructive. Back up any important cryptographic material on the partition before continuing. This destructiveness setting cannot be changed with a policy template (see "Guidelines and Restrictions" on page 96).

To manually set Key Export mode on a partition:

1. Launch LunaCM and log in to the partition as Partition SO.

   lunacm:>**slot set slot** <slotnum>

   lunacm:>**role login -name po**

2. Set **partition policy 0: Allow private key cloning** to **0** (OFF).

   lunacm:>**partition changepolicy -policy 0 -value 0**

3. Set **partition policy 1: Allow private key wrapping** to **1** (ON).

   lunacm:>**partition changepolicy -policy 1 -value 1**

To initialize a partition in Key Export mode using a policy template:

Use a standard text editor to include the following lines in the policy template file (see "Editing a Policy Template" on page 94):

```
0:"Allow private key cloning":0:1:0
1:"Allow private key wrapping":1:1:0
```

No Backup Mode

A partition in No Backup mode has the following restrictions:

- Private keys cannot be cloned to other partitions or to a SafeNet Luna Backup HSM. All other objects can still be cloned.
- Private keys cannot be wrapped off the HSM (exported to a file encrypted with a wrapping key). All other objects can still be wrapped off.

Without backup capability, private keys can never leave the HSM. This mode is useful when keys are intended to have short lifespans, and are easily replaced.

Setting No Backup Mode on a Partition

The Partition SO can use the following procedure to set No Backup mode. Use **partition showpolicies** to see the current policy settings.

To manually set No Backup mode on a partition:

1. Launch LunaCM and log in to the partition as Partition SO.

lunacm:>**slot set slot** <slotnum>

lunacm:>**role login -name po**

2. If **partition policy 0: Allow private key cloning** is set to **1** (ON), set it to **0** (OFF).

   lunacm:>**partition changepolicy -policy 0 -value 0**

3. If **partition policy 1: Allow private key wrapping** is set to **1** (ON), set it to **0** (OFF).

   lunacm:>**partition changepolicy -policy 1 -value 0**

To initialize a partition in No Backup mode using a policy template:

Use a standard text editor to include the following lines in the policy template file (see "Editing a Policy Template" on page 94):

```
0:"Allow private key cloning":0:1:0
1:"Allow private key wrapping":0:1:0
```

CHAPTER 10: Partitions

This chapter describes how to administer HSM administrative and application partitions on the HSM. It contains the following sections:

> "About HSM Partitions" below
> "Configured and Registered Client Using an HSM Partition" on the next page
> "Activation and Auto-Activation on PED-Authenticated Partitions" on page 178
> "Security of Your Partition Challenge" on page 182
> "Removing Partitions" on page 184
> "Frequently Asked Questions" on page 185

About HSM Partitions

HSM Partitions are independent logical HSMs that reside within the SafeNet Luna HSM inside, or attached to, your host computer or appliance. Each HSM Partition has its own data, access controls, security policies, and separate administration access, independent from other HSM partitions. HSM Partitions are analogous to 'safe deposit boxes' that reside within a bank's 'vault'. The HSM (vault) itself offers an extremely high level of security for all the contents inside. Each partition (safe deposit box) within the HSM also has its own security and access controls, so that even though the HSM security officer (bank manager) has access to the vault, they still cannot open the individual partitions (safe deposit boxes), because only the owner of the partition (safe deposit box) holds the key that opens it.

HSMs have two types of partitions:

> An administrative partition
> One or more application partitions

The Administrative Partition

Each HSM has a single administrative partition, which is created when the HSM is initialized. The administrative partition is owned by the HSM security officer (SO). This partition is used by the HSM SO and Auditor roles and is not normally used to store cryptographic objects.

Application Partitions

Application partitions are used to store the cryptographic objects used by your applications. Application partitions have their own partition SO, distinct from the HSM SO. For instructions on how to create application partitions, see "Creating an Application Partition on the HSM" on page 1 in the *Configuration Guide*.

The HSM SO is responsible for initializing the HSM, setting the HSM-wide policies, and creating empty application partitions. After the HSM SO creates the partition, complete control of the application partition is handed off to the partition SO. The HSM SO has no oversight over application partitions and can do nothing with them except delete them, if required.

The partition SO is responsible for setting the partition policies and for creating the Crypto Officer and optional Crypto User roles, who use the partition for cryptographic operations. Application partitions can be assigned to a single client, or multiple clients can be assigned to, and share, a single application partition.

Configured and Registered Client Using an HSM Partition

Following the instructions in the previous sections, you have already registered and assigned a Client to a SafeNet Luna PCIe HSM partition.

All that is required for a Client application to begin using a SafeNet Luna PCIe HSM partition (to which the Client has been assigned) is the standard handshake sequence:

1. The Client establishes a Network Trust Link connection with the SafeNet Luna PCIe HSM (port 1792).
2. The Client requests a list of available partitions (if not already known).
3. SafeNet Luna PCIe HSM responds with a list of only those partitions assigned to the requesting Client.
4. The Client chooses a partition from the available, assigned partitions.
5. SafeNet Luna PCIe HSM demands the credential (password or PED key) for the selected partition.
6. The Client (which may also be called Crypto User if you are using the Crypto Officer/Crypto User authentication and access model) provides the appropriate credential.
7. SafeNet Luna PCIe HSM grants access, and the Client application begins using the partition.

Your application should be capable of performing the above actions.

Simple Troubleshooting

If your Client application is having difficulty using SafeNet Luna PCIe HSM, and you have already verified the connection and the configuration (using multitoken and CMU utilities - see "Multitoken" on page 1 or "About the CMU Functions" on page 1 in the *Utilities Guide*), then there may be a problem with the configuration of your Client application. Try the following suggestions before calling Thales Group Technical Support.

Password Authentication Model

If you have a password-authenticated SafeNet Luna PCIe HSM, look to your application setup for the source of the problem. It might require special configuration. If SafeNet Luna PCIe HSM has replaced another HSM product (including a SafeNet product), you may need to modify the application to recognize the new device.

> **NOTE** Refer to the *SDK Reference Guide* and to the application integration documents provided by Thales GroupTechnical Support for information on integrating many popular applications and services with SafeNet Luna PCIe HSM.

PED Authentication Model

If you have a PED-authenticated SafeNet Luna PCIe HSM, having the Client application present the partition password is not sufficient to access the partition. The partition must also be activated (see "Activation and Auto-Activation on PED-Authenticated Partitions" on the next page). To ensure that the HSM Partition is always in the desired state, we recommend that you enable AutoActivation on the partition, so that it can accept Client authentication and access at any time without presenting a PED key at the SafeNet Luna PCIe HSM appliance.

If you want minute-by-minute control of a client's ability to access the HSM, without the need to access the appliance at its location, use the Remote PED feature (see "About Remote PED" on page 198).

Activation and Auto-Activation on PED-Authenticated Partitions

By default, PED-authenticated partitions require that a PED key and PED PIN be provided each time a user or application authenticates to the HSM. For some use cases, such as key vaulting, it may be desirable to require a physical key to access the HSM. For most application use cases, however, it is impractical to require this credential every time.

To address this limitation, you can enable **partition policy 22: Allow activation** on PED-authenticated HSM partitions. When partition policy 22 is enabled, the PED key secret for the CO or CU role is cached on the HSM the first time you authenticate. Clients can then connect to the partition without presenting the PED key. All that is required to authenticate is the PED challenge secret (password) for the activated role.

> **NOTE** Activation requires that a challenge secret is set for the role you want to activate. If the role does not have a challenge secret, you will continue to be prompted for the PED key, regardless of the policy setting.

Activation is not a big advantage for clients that connect and remain connected. It is an indispensable advantage in cases where clients repeatedly connect to perform a task and then disconnect or close the cryptographic session following completion of each task.

Tamper events and activation/auto-activation
When a tamper event occurs, or if an uncleared tamper event is detected on reboot, the cached PED key data is zeroized, and activation/auto-activation is disabled. See "Tamper Events" on page 281 and "Partition Capabilities and Policies" on page 87 for more information.

Enabling Activation on a Partition

Activation is controlled by **partition policy 22: Allow activation**. The Partition SO can set this policy in LunaCM, using the **partition changepolicy** command. When partition policy 22 is enabled, the HSM checks for the following conditions each time the Crypto Officer (CO) or Crypto User (CU) perform an action that requires authentication:

- Is PED key secret for the role cached on the HSM?
- Has a challenge secret been created for the role?

The HSM responds as follows:

- If the PED key secret is not currently cached, you are prompted for the PED key. The PED key secret is cached when you provide the PED key.
- If the PED key secret is already cached, but a challenge secret has not been created for the role, you are prompted for the PED key.

After the role is activated and a challenge secret is set, the PED key is no longer required for that role to login to the partition, and it can be stored safely. The CO or CU can connect to the partition and perform role-specific operations from any registered client, using only the PED challenge password.

To enable activation on an application partition:

1. Log in to the partition as the Partition SO.

   lunacm:>**role login -name Partition SO**

2. Enable **partition policy 22: Allow activation**.

   lunacm:>**partition changepolicy -slot** <slot number> **-policy 22 -value 1**

Activating a Role

After enabling partition policy 22, activate the CO and/or CU roles on the partition. You must set a PED challenge password for each role you want to activate. The Partition SO must set the initial challenge secret for the Crypto Officer, who must set it for the Crypto User. The role will become activated the first time the role logs in to the partition.

To activate a role (Partition SO):

1. Ensure that **partition policy 22: Allow activation** is enabled (set to 1):

   lunacm:>**partition showpolicies**

   If it is not set, log in as the Partition SO and use the **partition changepolicy** command to enable the policy, as described in "Enabling Activation on a Partition" on the previous page.

2. Create an initial challenge secret for the Crypto Officer.

   lunacm:>**role createchallenge -name co**

```
lunacm:>role createchallenge -name co

        Please attend to the PED.

        enter new challenge secret: ********

        re-enter new challenge secret: ********

Command Result : No Error
```

3. Provide the initial challenge secret to the Crypto Officer by secure means. The CO will need to change the challenge secret before using the partition for any crypto operations.

4. Log out as Partition SO.

   lunacm:>**role logout**

To activate a role (Crypto Officer)

1. Login as Crypto Officer (or enter any command that requires authentication).

   lunacm:>**role login -name co**

```
lunacm:>role login -n co

        enter password: ********

        Please attend to the PED.

Command Result : No Error
```

   The Crypto Officer PED secret is cached, and the role is now activated.

2. If you have not already done so on a previous login, change the initial CO PED secret. By default, the PED secret provided by the Partition SO expires after the initial login. If **HSM policy 21: Force user PIN change after set/reset** is set to **0** (off), you can continue to use the PED secret provided.

   lunacm:>**role changepw -name co**

```
lunacm:> role changepw -name co

        This role has secondary credentials.
        You are about to change the primary credentials.
        Are you sure you wish to continue?

        Type 'proceed' to continue, or 'quit' to quit now -> proceed


Command Result : No Error
```

3. Change the initial CO challenge secret. You must include the **-oldpw** option to indicate that you wish to change the challenge secret (referred to as the secondary credential), rather than the black PED key (primary credential).

   lunacm:>**role changepw -name co -oldpw** <initial_challenge> **-newpw** <new_challenge>

```
lunacm:>role changepw -name co -oldpw password -newpw Pa$$w0rd

        This role has secondary credentials.
        You are about to change the secondary credentials.
        Are you sure you wish to continue?

        Type 'proceed' to continue, or 'quit' to quit now ->proceed

        Please attend to the PED.

Command Result : No Error
```

4. [Optional] Create an initial challenge secret for the Crypto User.

   lunacm:>**role createchallenge -name cu**

```
lunacm:>role createchallenge -name cu

        Please attend to the PED.

        enter new challenge secret: ********

        re-enter new challenge secret: ********

Command Result : No Error
```

5. [Optional] Provide the initial challenge secret to the Crypto User by secure means. The CU will need to change the challenge secret before using the partition for any crypto operations.

6. Log out as Crypto Officer.

   lunacm:>**role logout**

With activation in place, you can log in once and put your black CO PED key away in a safe place. The cached credentials will allow your application(s) to open and close sessions and perform their operations within those sessions.

To activate a role (Crypto User)

1. Login to the partition as the Crypto User. When prompted, enter the initial challenge secret.

   lunacm:>**role login -name cu**

```
lunacm:>role login -n cu

        enter password: ********

        Please attend to the PED.

Command Result : No Error
```

2. If you have not already done so on a previous login, change the initial CU PED secret. By default, the PED secret provided by the Crypto Officer expires after the initial login. If **HSM policy 21: Force user PIN change after set/reset** is set to **0** (off), you can continue to use the PED secret provided.

   lunacm:>**role changepw -name cu**

```
lunacm:> role changepw -name cu

        This role has secondary credentials.
        You are about to change the primary credentials.
        Are you sure you wish to continue?

        Type 'proceed' to continue, or 'quit' to quit now -> proceed


Command Result : No Error
```

3. Change the initial CU challenge secret. You must include the **-oldpw** option to indicate that you wish to change the challenge secret (referred to as the secondary credential), rather than the gray PED key (primary credential).

   lunacm:>**role changepw -name cu -oldpw** <initial_challenge> **-newpw** <new_challenge>

```
lunacm:>role changepw -name cu -oldpw password -newpw Pa$$w0rd

        This role has secondary credentials.
        You are about to change the secondary credentials.
        Are you sure you wish to continue?

        Type 'proceed' to continue, or 'quit' to quit now ->proceed

        Please attend to the PED.

Command Result : No Error
```

With activation in place, you can log in once and put your gray CO PED key away in a safe place. The cached credentials will allow your application(s) to open and close sessions and perform their operations within those sessions.

Deactivating a Role on an Activated Partition

An activated role on a partition remains activated until one of the following actions occurs:

- You explicitly deactivate the role using the LunaCM **role deactivate** command. The role is deactivated until the next time you perform an action (such as **role login**) that requires authentication for the role, at which time the authentication credential is re-cached.

> Power is lost to the HSM. You can use auto-activation to automatically reactivate a partition after a short power loss, if desired. See "Auto-Activation" below.

To deactivate a role on a partition (Partition SO)

1. Enter the following command to deactivate an activated role on a partition:

   lunacm:> **role deactivate -name** <role>

   This deletes the cached authentication credential for the role. The next time a login or activation is performed, the credential is re-cached.

2. If you wish to disable activation entirely, so that credentials are not re-cached at the next login, the Partition SO can disable **partition policy 22: Allow activation**.

   lunacm:>**partition changepolicy -policy 22 -value 0**

3. If partition policy 22 is disabled, auto-activation is also disabled (even though **partition policy 23: Allow auto-activation** is set to **1**). When partition policy 22 is enabled again, auto-activation resumes. To turn off auto-activation, you must disable partition policy 23.

   lunacm:>**partition changepolicy -policy 23 -value 0**

Auto-Activation

Auto-activation enables PED key credentials to be cached even in the event of a restart or a short power outage (up to 2 hours). Clients can re-connect and continue using the application partition without needing to re-authenticate using a PED key.

The ability to auto-activate a partition is controlled by **partition policy 23: Allow auto-activation**. To enable auto-activation, the Partition SO can use the LunaCM **partition changepolicy** command to set partition policy 23 to **1**.

When partition policy 23 is enabled, auto-activation is set for the partition the first time an activated role (CO or CU) logs in. If the authentication data requires refreshing, the PED prompts you for the appropriate black or gray PED key and PIN. Once login is complete, the PED credential is cached, and the client can begin using the activated application partition.

To auto-activate an application partition (Partition SO)

1. Ensure that **partition policy 22: Allow activation** is enabled.
2. Login to the partition as Partition SO.

   lunacm:>**role login -name po**

3. Set **partition policy 23: Allow auto-activation** to **1**.

   lunacm:>**partition changepolicy -policy 23 -value 1**

   Auto-activation will begin for each affected role (CO or CU) the next time the role is authenticated.

Security of Your Partition Challenge

For SafeNet Luna PCIe HSMs with Password Authentication, the partition password used for administrative access by the Crypto Officer is also the partition challenge secret or password used by client applications.

For SafeNet Luna PCIe HSMs with PED Authentication, the partition authentication used for administrative access by the Crypto Officer is the secret on the black PED key(s) for that partition. The partition challenge secret or password used by client applications is a separate character string, set by the Partition SO and then changed by the Crypto Officer (mandatory) for the CO's use. This is one way in which we implement separation of roles in the SafeNet Luna HSM security paradigm.

How Secure Is the Challenge Secret or Password?

The underlying concern is that a password-harvesting attack might eventually crack the secret that protects the partition. Layers of protection are in place, to minimize or eliminate such a risk.

First, such an attack must be run from a SafeNet Luna Client computer. For interaction with HSM partitions on a SafeNet network appliance, like SafeNet Luna Network HSM, a SafeNet Luna Client computer is one with SafeNet software installed, on which you have performed the exchange of certificates to create a Network Trust Link (NTL). That exchange requires the knowledge and participation of the appliance administrator and the Partition SO (who might, or might not, be the same person). It is not possible to secretly turn a computer into a Client of a SafeNet Luna HSM partition - an authorized person within your organization must participate.

Second, for SafeNet Luna HSMs with password authentication, you set the partition password directly when you create the partition, so you can make it as secure as you wish (for an example of guidance on password strength, see http://howsecureismypassword.net/ or http://xkcd.com/936/)

For SafeNet Luna HSMs with PED authentication, an optional partition password (also called a challenge secret) may be added for the initialized Crypto Officer (CO) and/or Crypto User (CU) roles. See "role createchallenge" on page 1 of the *LunaCM Command Reference Guide* for the proper command syntax.

Using LunaCM or LunaSH, you can change the partition password (or challenge secret) if you suspect it has been compromised, or if you are complying with a security policy that dictates regular password changes.

As long as you replace any password/challenge secret with one that is equally secure, the possible vulnerability is extremely small.

Conversely, you can choose to replace a secure, random password/challenge-secret with one that is shorter or more memorable, but less secure - you assume the risks inherent in such a tradeoff.

Third, SafeNet Luna HSM **partition policy 15: Ignore failed challenge responses** can be set to **0** (off). When that policy is off, the HSM stops ignoring bad challenge responses (that is, attempts to submit the partition secret) and begins treating them as failed login attempts. Each bad login attempt is counted. **Partition policy 20: Max failed user logins allowed** determines how high that count can go before the partition is locked out.

Once a partition is locked by bad login attempts, it cannot be accessed until the HSM Security Officer (SO) unlocks it. This defeats an automated harvesting attack that relies on millions of attempts occurring at computer-generated speeds. As well, after one or two lockout cycles, the HSM SO would realize that an attack was under way and would rescind the NTL registration of the attacking computer. That computer would no longer exist as far as the HSM partition was concerned. The SO or your security organization would then investigate how the client computer had been compromised, and would correct the problem before allowing any new NTL registration from that source. See "Failed Login Attempts" on page 329 for more information.

As the owner/administrator of the HSM, you determine any tradeoffs with respect to security, convenience, and other operational parameters.

Removing Partitions

Only the HSM Security Officer can remove HSM partitions. When a partition is removed, it is cleared from the HSM and all of its contents are deleted.

To remove a partition from the HSM:

1. Open LunaCM (or run **slot list** if it is already open) and note the slot number of the partition you wish to remove.

```
LunaCM v7.0.0. Copyright (c) 2006-2017 SafeNet.

        Available HSMs:

        Slot Id ->              3
        Label ->                myLunapar
        Serial Number ->        154438865287
        Model ->                Luna K7
        Firmware Version ->     7.0.1
        Configuration ->        Luna User Partition With SO (PW) Signing With Cloning Mode
        Slot Description ->     User Token Slot

        Slot Id ->              4
        Label ->
        Serial Number ->        154438865291
        Model ->                Luna K7
        Firmware Version ->     7.0.1
        Configuration ->        Luna User Partition With SO (PW) Signing With Cloning Mode
        Slot Description ->     User Token Slot

        Slot Id ->              103
        Label ->                myPCIeHSM
        Serial Number ->        66331
        Model ->                Luna K7
        Firmware Version ->     7.0.1
        Configuration ->        Luna HSM Admin Partition (PW) Signing With Cloning Mode
        Slot Description ->     Admin Token Slot
        HSM Configuration ->    Luna HSM Admin Partition (PW)
        HSM Status ->           L3 Device

        Current Slot Id: 3
```

2. Set the active slot to the HSM Admin partition and login as HSM SO.

   lunacm:>**slot set slot** <slotnum>

   lunacm:>**role login -name so**

3. Delete the partition by specifying its slot number.

   lunacm:>**partition delete -slot** <slotnum>

```
lunacm:>partition delete -slot 4

        You are about to delete partition.
        Are you sure you wish to continue?

        Type 'proceed' to continue, or 'quit' to quit now ->proceed

Command Result : No Error
```

Frequently Asked Questions

This section provides additional information by answering questions that are frequently asked by our customers.

Why do I get an error when I attempt to set the partition policies for activation (22) and auto-activation (23) on my password authenticated SafeNet Luna Network HSM?
Those policies apply to PED-authenticated SafeNet Luna Network HSM, only.
For both PED-authenticated and password-authenticated HSMs, your client authenticates to a partition with a challenge password.

For PED-authenticated HSMs, the application partition must be in a state where it is able to accept that challenge password. The extra layer of authentication - the partition Crypto Officer's black PED key or the Crypto User's gray PED key - must have been presented first before the partition can be receptive to the challenge/password.

Password-authenticated HSMs have only the single layer of authentication - the challenge/password is all that is needed. The password is both the client authentication and the partition administrator (Crypto Officer/Crypto User) authentication.

For PED-authenticated HSMs, Activation and Auto-Activation enable caching of the first layer of authentication to provide a level of operational convenience similar to that of the password-authenticated HSMs.

So, what is the difference in security, once Activation and Auto-Activation are started?
From the convenience point of view, none. But, whereas the password-authenticated partition is "open for business" to anybody with that partition's password, as soon as the partition is created, a PED-authenticated partition is not. One implication is that all partitions of a multi-partition password-authenticated HSM are available whenever any of them are available, which is essentially whenever the HSM is powered on.

The owner of a PED-authenticated HSM partition can disable client access to just one partition by deactivating (de-caching) just that one partition's PED key authentication, so that the challenge/password is not accepted. Any other partitions on that HSM that are not deactivated (i.e., still have their black PED key or gray PED key authentication cached) are still able to accept challenge/password from their clients.

You are not required to cache the PED key data in order to use a partition. You could, if you preferred, simply leave the PED key for that partition inserted in a connected Luna PED, and press keypad keys on the PED whenever first-level authentication for partition access was required. Since this would defeat much of the reason for having a powerful networked HSM server, generally nobody does this with SafeNet Luna Network HSM in a production environment. As well, if you have created both a Crypto Officer and a Crypto User for your partition, you would need to switch out the black PED key or the gray PED key, whenever the other entity needed to PED-authenticate while the PED key authentications are not cached.

You also have the option of partially engaging the PED key caching feature by enabling Activation without enabling Auto-activation. In that case, you present your PED key to activate the partition - which allows it to accept its partition challenge/password from clients - and the cached black PED key or gray PED key authentication data is retained while the HSM has power (or until you explicitly de-cache). But the cached authentication does not survive a power outage or an intentional power cycle (because you chose to Activate, but not to autoActivate as well). Thus, by applying different policy settings, you could have some partitions on

your PED-authenticated HSM able to return to client availability immediately following a power-cycle/outage (no human intervention needed), while others would wait for your intervention, with a black PED key (Crypto Officer) or a gray PED key (Crypto User), before becoming client-available.

Finally, Activation and Auto-Activation are partition-level policy settings, not role-level. Therefore, if the policy is on, it is on for all roles. If the policy is off, it is off for all roles. You cannot individually cache authentication data from a gray PED key, but not from a black PED key (or the opposite) within a single partition.

CHAPTER 11: PED Authentication

The SafeNet Luna PIN Entry Device (Luna PED) provides PIN entry and secret authentication to a SafeNet Luna HSM that requires Trusted Path Authentication. The requirement for PED or password authentication is configured at the factory, according to the HSM model you selected at time of purchase.

The Luna PED and PED keys are the only means of accessing the PED-authenticated HSM's administrative functions. They prevent key-logging exploits on workstations connected to the host HSM, because authentication is delivered directly from the hand-held PED to the HSM via the independent, trusted-path interface. No password is entered via computer keyboard.

SafeNet Luna PCIe HSM release 7.x requires Luna PED firmware version 2.7.1 or higher. This firmware is backward-compatible with legacy SafeNet Luna PCIe HSM 6.x.

This chapter contains the following sections about PED authentication:

> "SafeNet Luna PED Hardware Functions" on page 193
> "Local PED Setup" on page 196
> "About Remote PED" on page 198
> "Remote PED Setup" on page 202
  - "Initializing the Remote PED Vector (RPV) and Creating the Orange PED Key" on page 202
  - "Installing PEDserver and Setting Up the Remote Luna PED" on page 203
  - "Opening a Remote PED Connection" on page 205
  - "Ending or Switching the Remote PED Connection" on page 207
  - "Remote PED Troubleshooting" on page 208
> "PED Key Management" on page 211
  - "Creating PED Keys" on page 212
  - "Performing PED Authentication" on page 217
  - "Consequences of Losing PED Keys" on page 218
  - "Identifying a PED Key Secret" on page 221
  - "Duplicating Existing PED Keys" on page 222
  - "Changing a PED Key Secret" on page 222
> "PEDserver and PEDclient" on page 225

PED Authentication Architecture

The PED Authentication architecture consists of the following components:

- **SafeNet Luna PED:** a PIN Entry Device with a local or remote connection to the HSM. The PED reads authentication secrets from PED keys on behalf of an HSM or partition (see "SafeNet Luna PED Hardware Functions" on page 193).
- **Authentication secrets:** Cryptographic secrets generated by the HSM and stored on PED keys. These secrets serve as login credentials for the various roles on the HSM. They can be shared among roles, HSMs, and partitions according to your security scheme.
- **PED Keys:** physical USB-connected devices that contain authentication secrets, created by the HSM (see "PED Keys" on the next page). PED Keys have the following custom authentication features:
  - **Shared Secrets:** PED keys of the same type can be reused or shared among HSMs or partitions, allowing domain sharing (necessary for HA and backup configurations), legacy-style Security Officer authentication, and other custom configurations. See "Shared PED Key Secrets" on page 191.
  - **PED PINs:** optional PINs associated with specific PED keys, set by the owner of the PED key at the time of creation. PED PINs offer an extra layer of security for PED keys which could be lost or stolen. See "PED PINs" on page 192.
  - **M of N Split Key Scheme:** optional configuration which allows a role to split its authentication secret across multiple PED keys, and require a minimum number of those keys for authentication. This scheme can be customized to be as simple or complex as your organization's security policy dictates. See "M of N Split Secrets (Quorum)" on page 192.

Comparing Password and PED Authentication

The following table describes key differences between password- and PED-authenticated HSMs.

	Password-authentication	PED-authentication
Ability to restrict access to cryptographic keys	> Knowledge of role password is sufficient > For backup/restore, knowledge of partition domain password is sufficient	> Ownership of the black Crypto Officer PED key is mandatory > For backup/restore, ownership of both black CO and red domain PED keys is mandatory > The Crypto User role is available to restrict access to read-only, with no key management authority > Option to associate a PED PIN with any PED key, imposing a two-factor authentication requirement on any role
Dual Control	> Not available	> MofN (split-knowledge secret sharing) requires "M" different holders of portions of the role secret (a quorum) in order to authenticate to an HSM role - can be applied to any, all, or none of the administrative and management operations required on the HSM
Key-custodian responsibility	> Password knowledge only	> Linked to partition password knowledge > Linked to black PED key(s) ownership and optional PED PIN knowledge

	Password-authentication	PED-authentication
Two-factor authentication for remote access	> Not available	> Remote PED and orange (Remote PED Vector) PED key deliver highly secure remote management of HSM, including remote backup

PED Keys

A PED key is a USB authentication device, embedded in a molded plastic body. It contains a secret, generated by the HSM, that authenticates a role, cloning domain, or remote PED server. This secret is retained until deliberately changed by an authorized user.

The Luna PED does not hold the authentication secrets. They reside only on the portable PED keys.

PED keys are created when an HSM, partition, role, or Remote PED vector is initialized. A PED key can contain only one authentication secret at a time, but it can be overwritten with a new authentication secret. See "PED Key Management" on page 211.

> **CAUTION!** Do not subject PED keys to extremes of temperature, humidity, dust, or vibration. Use the included key cap to protect the USB connector.

PED Key Types and Roles

The PED uses PED keys for all credentials. You can apply the appropriate labels included with your PED keys, according to the table below, as you create them.

The PED key colors correspond with the HSM roles described in "HSM Roles and Procedures" on page 322. The following table describes the keys associated with the various roles:

Lifecycle	PED Key	PED Secret	Function
HSM Administration	**Blue**	HSM Security Officer (HSM SO) secret	Authenticates the HSM SO role. The HSM SO manages provisioning functions and security policies for the HSM. **Mandatory**
	Red 	HSM Domain or Key Cloning Vector	Cryptographically defines the set of HSMs that can participate in cloning for backup. See "Domain PED Keys" on the next page. **Mandatory**
	Orange 	Remote PED Vector	Establishes a connection to a Remote PED server. **Optional**
HSM Auditing	**White** 	Auditor (AU) secret	Authenticates the Auditor role, responsible for audit log management. This role has no access to other HSM services. **Optional**
Partition Administration	**Blue**	Partition Security Officer (PO) secret	Authenticates the Partition SO role. The PO manages provisioning activities and security policies for the partition. **NOTE:** If you want the HSM SO to also perform Partition SO duties, you can use the same blue key to initialize both roles. **Mandatory**
	Red 	Partition Domain or Key Cloning Vector	Cryptographically defines the set of partitions that can participate in cloning for backup or high-availability. See "Domain PED Keys" on the next page. **Mandatory**

Lifecycle	PED Key	PED Secret	Function
Partition Operation	**Black** 	Crypto Officer (CO) secret	Authenticates the Crypto Officer role. The CO can perform both cryptographic services and key management functions on keys within the partition. **Mandatory**
	Gray 	Crypto User (CU) secret	Authenticates the Crypto User role. The CU can perform cryptographic services using keys already existing within the partition. It can create and back up public objects only. **NOTE:** If administrative separation is not important, you can use a single black key to initialize the Crypto Officer and Crypto User roles and still have two separate challenge secrets to distinguish read-write and read-only role privileges. **Optional**

Shared PED Key Secrets

The Luna PED identifies the type of authentication secret on an inserted PED key, and secrets of the same type (color designation) can be used interchangeably. During the key creation process, you have the option of reusing an authentication secret from an existing key rather than have the HSM create a new one. This means that you can use the same PED key(s) to authenticate multiple HSMs or partitions. This is useful for:

- legacy-style authentication schemes, where the HSM SO also functions as the owner of application partitions. This is achieved by using the same blue PED key to initialize the HSM and some or all of the partitions on the HSM.
- allowing a single HSM SO to manage multiple HSMs, or a single Partition SO to manage multiple partitions
- ensuring that HSMs/partitions share a cloning domain (see "Domain PED Keys" below)
- allowing a read-write Crypto Officer role and a read-only Crypto User role to be managed by the same user

It is not necessary for partitions in an HA group to share the same blue Partition SO key. Only the red cloning domain key must be identical between HA group members.

> **NOTE** Using a single PED key secret to authenticate multiple roles, HSMs, or partitions is less secure than giving each its own PED key. Refer to your organization's security policy for guidance.

Domain PED Keys

A red domain PED key holds the key-cloning vector (the domain identifier) that allows key cloning between HSMs and partitions, and is therefore the PED key most commonly shared between HSMs or partitions. Cloning is a secure method of copying cryptographic objects between HSMs and partitions, required for backup/restore and within HA groups. It ensures that keys copied between HSMs or partitions are:

- strongly encrypted
- copied only between HSMs and partitions that share a cloning domain.

NOTE An HSM or partition can be a member of only one domain, decided at initialization. A domain can only be changed by re-initializing the HSM. Partition domains may not be changed after initialization.

PED PINs

The Luna PED allows the holder of a PED key to set a numeric PIN, 4-48 characters long, to be associated with that PED key. This PIN must then be entered on the PED keypad for all future authentication. The PED PIN provides two-factor authentication and ensures security in case a key is lost or stolen. If you forget your PED PIN, it is the same as losing the PED key entirely; you cannot authenticate the role.

PED PINs can be set only at the time of key creation, and can be changed only by changing the secret on the PED key. Duplicate keys made at the time of creation can have different PED PINs, allowing multiple people access to the role (see "Creating PED Keys" on page 212). Copies made later are true copies with the same PED PIN, intended as backups for one person (see "Duplicating Existing PED Keys" on page 222). Duplicates of the PED key all have the same PED PIN.

If you are using an M of N configuration, each member of the M of N keyset may set a different PED PIN.

CAUTION! Forgetting a PED PIN is equivalent to losing the key entirely; you can no longer authenticate the role, domain, or RPV. See "PED Authentication" on page 187.

M of N Split Secrets (Quorum)

The Luna PED can split an authentication secret among multiple PED keys (up to 16), and require a minimum number of the split keys (a quorum of key-holders) to authenticate the role. This provides a customizable layer of security by requiring multiple trusted people (sometimes called the quorum) to be present for authentication to the role.

For example, you could decide (or your security policy could dictate) that at least three trusted people must be present for changes to the HSM policies or for client partition assignments. To accommodate illness, vacations, business travel, or any other reasons that a key-holder might not be present at the HSM site, it is advisable to split the authentication secret between more than three people. If you decide on a five-key split, you would specify M of N for the HSM SO role to be 3 of 5. That is, the pool of individual holders of spits of that role secret is five persons, and from among them, a quorum of three must be available to achieve authentication.

In this scenario, the HSM SO authentication secret is split among five blue PED keys, and at least three of those keys must be presented to the Luna PED to log in as HSM SO.

This feature can be used to customize the level of security and oversight for all actions requiring PED authentication. You can elect to apply an M of N split-secret scheme to all roles and secrets, to some of them, or to none of them. If you do choose to use M of N, you can set different M and N values for each role or secret. Please note the following recommendations:

- M = N is not recommended; if one of the key holders is unavailable, you cannot authenticate the role.
- M = 1 is not recommended; it is no more secure than if there were no splits of the secret - a single person can unlock the role without oversight. If you want multiple people to have access to the role, it is simpler to create multiple copies of the PED key.

> **NOTE** Using an M of N split secret can greatly increase the number of PED keys you require. Ensure that you have enough blank or rewritable PED keys on hand before you begin backing up your M of N scheme.

Activated Partitions and M of N

For security reasons, the HSM and its servers are often kept in a locked facility, and accessed under specific circumstances, directly or by secure remote channel. To accommodate these security requirements, the Crypto Officer and Crypto User roles can be Activated (to use a secondary, alpha-numeric login credential to authenticate), allowing applications to perform cryptographic functions without having to present a black or gray PED key (see "Activation and Auto-Activation on PED-Authenticated Partitions" on page 178). In this case, if the HSM is rebooted for maintenance or loses power due to an outage, the cached PED secret is erased and the role must be reactivated (by logging in the role via LunaCM and presenting the requisite M number or quorum of PED keys) before normal operations can resume.

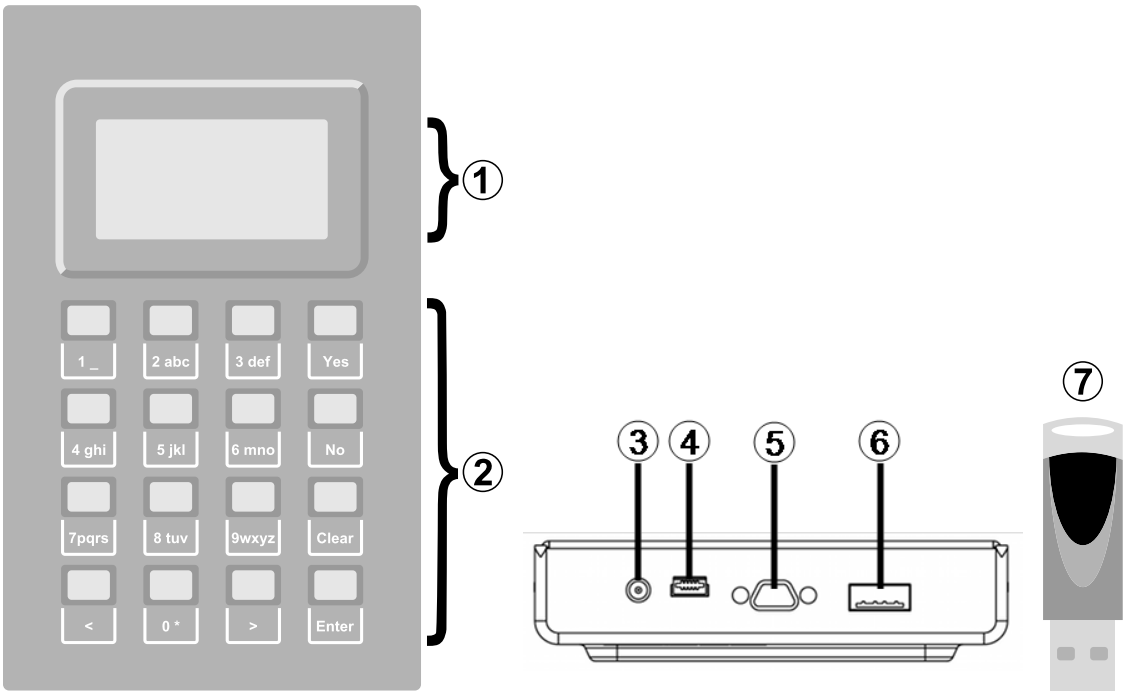
SafeNet Luna PED Hardware Functions

The SafeNet Luna PED reads authentication secrets from PED keys on behalf of an HSM or partition. This section contains the following information about the Luna PED device:

- "Physical Features" below
- "Keypad Functions" on the next page
- "Modes of Operation" on page 195
- "Admin Mode Functions" on page 196

Physical Features

The SafeNet Luna PED is illustrated below, with important features labeled.



1	Liquid Crystal Display (LCD), 8 lines.
2	Keypad for command and data entry. See "Keypad Functions" below.
3	DC power connector. Not used for PED version 2.8 and above.*
4	USB mini-B connector. Used for connecting to the HSM and for file transfer to or from the PED. PED version 2.8 and above is powered by this USB connection.
5	Micro-D subminiature (MDSM) connector. Not used for Luna release 7.x.
6	USB A-type connector for PED keys.
7	PED key. Keys are inserted in the PED key connector (item 6).

* PEDs with firmware version 2.8 and above are powered by any USB 2.x or 3.x connection, and do not have an external DC power supply. The PED driver must be installed on the connected computer. If the PED is connected to a hub or to a computer without the driver, then the PED display backlight illuminates, but no PED menu is presented.)

Keypad Functions

The Luna PED keypad functions are as follows:

Key	Function
Clear	> Clear the current entry, such as when entering a PED PIN > Hold the key down for five seconds to reset the PED during an operation. This applies only if the PED is engaged in an operation or is prompting for action. There is no effect when no command has been issued or when a menu is open
<	> **Backspace:** clear the most recent digit you typed on the PED > **Exit:** return to the previous PED menu
>	> **Log:** displays the most recent PED actions (since entering Local or Remote Mode)
Numeric keys	> Select numbered menu items > Input PED PINs
Yes and No	> Respond to Yes or No questions from the PED
Enter	> Confirm an action or entry

Modes of Operation

The Luna PED can operate in four different modes, depending on the type of HSM connection you want to use:

> **Local PED-SCP:** This mode is reserved for legacy SafeNet Luna 6.x HSMs that use an MDSM connector between the PED and the HSM. It does not apply to Luna 7.x. Initial HSM configuration must be done in Local PED mode. See "Local PED Setup" on the next page for instructions.

> **Admin:** This mode is for upgrading the PED device firmware, diagnostic tests, and PED key duplication. See "Admin Mode Functions" on the next page for the functions available in this mode.

> **Remote PED:** In this mode, the PED is connected to a remote workstation and authenticated to the HSM with an orange PED key containing a Remote PED Vector (RPV) secret. This mode allows the SafeNet Luna PCIe HSM to be located in a data center or other location restricting physical access. See for more information.

> **Local PED-USB:** In this mode, the PED is connected directly to the HSM card with a USB mini-B to USB-A connector cable. Initial HSM configuration must be done in Local PED mode.

If the Luna PED is connected to an interface when it is powered up, it automatically detects the type of connection being used and switches to the appropriate mode upon receiving the first command from the HSM.

Changing Modes

If you change your PED configuration without disconnecting the PED from power, you must select the correct mode from the main menu.

To change the Luna PED's active mode

1. Press the **<** key to navigate to the main menu.

```
Select Mode
1  Local PED-SCP
4  Admin
7  Remote PED
0  Local PED-USB

      PED V.2.7.1-5
```

The main menu displays all the available modes, as well as the PED's current firmware version.

2. Press the corresponding number on the keypad for the desired mode.

> **NOTE** The Luna PED must be in **Local PED-USB** mode when connected to a Release 7.x SafeNet Luna PCIe HSM card, or LunaCM will return an error (CKR_DEVICE_ERROR) when you attempt authentication.

Admin Mode Functions

In this mode, you can upgrade the PED device software, run diagnostic tests, and duplicate PED keys without having the Luna PED connected to an HSM. Press the corresponding number key to select the desired function.

```
Admin mode...
1  PED Key
5  Backup Devices
7  Software Update
9  Self Test

<  EXIT
```

- **PED Key:** allows you to identify the secret on an inserted PED key, or duplicate the key, without having the Luna PED connected to an HSM.
- **Backup Devices:** Not applicable to Luna 7.x.
- **Software Update:** requires a PED software file and instructions sent from Thales Group.
- **Self Test:** test the PED's functionality. Follow the on-screen instructions to test button functions, display, cable connections, and the ability to read PED keys. The PED returns a PASS/FAIL report once it concludes the test.

Local PED Setup

A Local PED connection is the simplest way to set up the SafeNet Luna PED. In this configuration, the PED is connected directly to the HSM card. It is best suited for situations where all parties who need to authenticate credentials have convenient physical access to the HSM. When the HSM is stored in a secure data center and accessed remotely, you must use a Remote PED setup.

Setting Up a Local PED Connection

The SafeNet Luna PCIe HSM administrator can use these directions to set up a Local PED connection. You require:

- SafeNet Luna PED with firmware 2.7.1 or newer

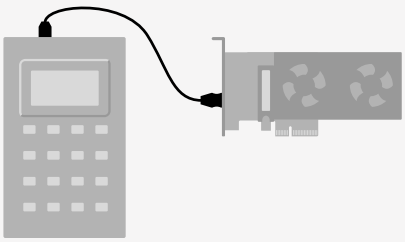
> USB mini-B to USB-A connector cable

> Luna PED DC power supply (if included with your Luna PED)

To set up a Local PED connection

1. Connect the Luna PED to the HSM using the supplied USB mini-B to USB-A connector cable.

   NOTE To operate in Local PED-USB mode, the PED must be connected directly to the HSM card's USB port, and not one of the other USB connection ports on the host system.

2. PED version 2.8 and above is powered via the USB connection. If you are using PED version 2.7.1, connect it to power using the Luna PED DC power supply.

   As soon as the PED receives power, it performs start-up and self-test routines. It verifies the connection type and automatically switches to the appropriate operation mode when it receives the first command from the HSM.

3. If you prefer to set the operation mode to **Local PED-USB** manually, see "Changing Modes" on page 195.

The Luna PED is now ready to perform authentication for the HSM. You may proceed with setting up or deploying your SafeNet Luna PCIe HSM. All commands requiring authentication (HSM/partition initialization, login, etc.) will now prompt the user for action on the locally-connected Luna PED.

PED Actions

There are several things that you can do with the Luna PED at this point:

> Wait for a PED authentication prompt in response to a LunaCM command (see "Performing PED Authentication" on page 217)

> Create copies of your PED keys (see "Duplicating Existing PED Keys" on page 222)

> Change to the Admin Mode to run tests or update PED software (see "Changing Modes" on page 195)

> Prepare to set up a Remote PED server (see "About Remote PED" on the next page)

Local PED Troubleshooting

If you encounter problems with Local PED, refer to this section.

CKR_PED_UNPLUGGED error after hsm restart

After running **hsm restart**, LunaCM returns a CKR_PED_UNPLUGGED error when authentication is attempted.

```
lunacm:>role login -n so

        Please attend to the PED.
```

```
Caution: You have only 3 so login attempts left. If you fail 3
         more consecutive login attempts (i.e. with no successful
         logins in between) the HSM will be ZEROIZED!!!

Error in execution: CKR_PED_UNPLUGGED.

Command Result : 0x8000002e (CKR_PED_UNPLUGGED)
```

If you receive this error, disconnect the Luna PED from the HSM's USB port and reconnect it before issuing the login command again.

About Remote PED

A Remote PED connection allows you to access PED-authenticated HSMs that are kept in a secure data center or other remote location where physical access is restricted or inconvenient. This section provides descriptions of the following aspects of Remote PED connections:

> "Remote PED Architecture" below
> "Remote PED Connections" on the next page
> "PEDserver-PEDclient Communications" on page 201

Remote PED Architecture

The Remote PED architecture consists of the following components:

> **Remote PED:** a Luna PED with firmware 2.7.1 or newer, connected to a network-connected workstation, powered on, and set to Remote PED mode.

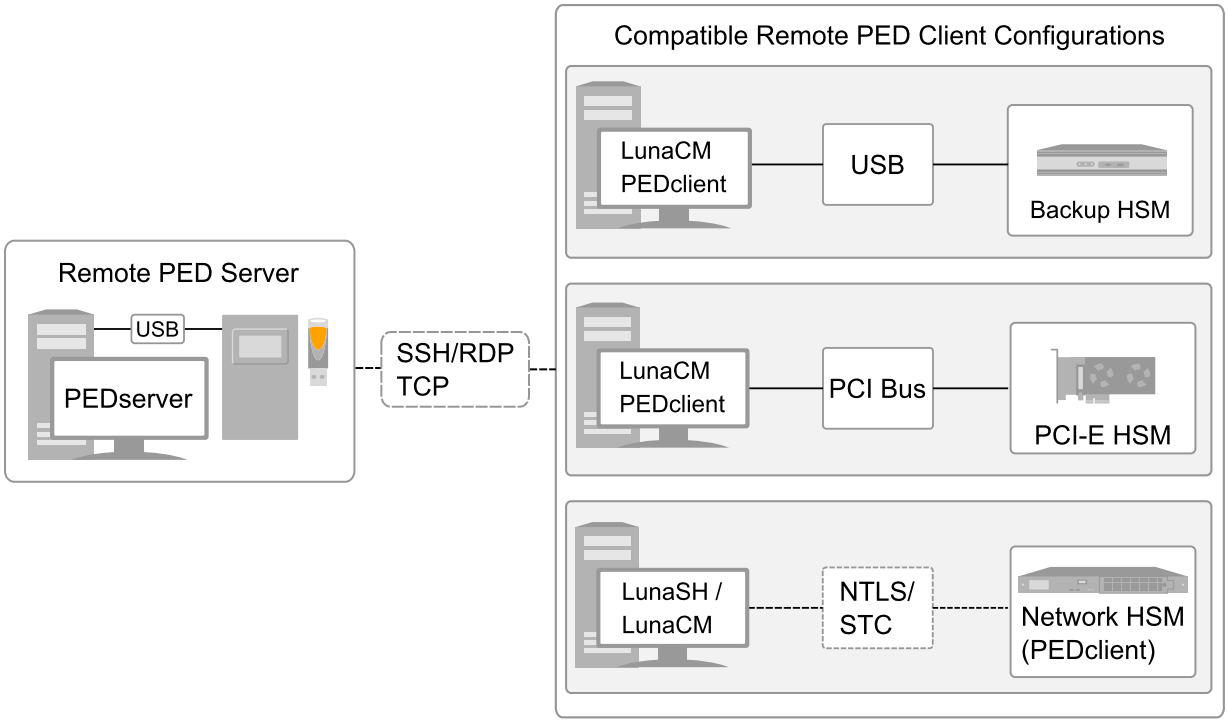
> **Remote PED Vector (RPV):** a randomly generated, encrypted value used to authenticate between a Remote PED (via PEDserver) and a SafeNet Luna HSM (via PEDclient).

> **Remote PED Key (RPK):** an orange PED key containing an RPV (or multiple PED keys with a split RPV in an M of N implementation).

> **PEDserver:** software that runs on the remote workstation with a USB-connected Luna PED. PEDserver accepts requests from and serves PED actions and data to PEDclient.

> **PEDclient:** software that requests remote PED services from PEDserver. PEDclient runs on the network-connected system hosting the HSM, which can be one of the following:

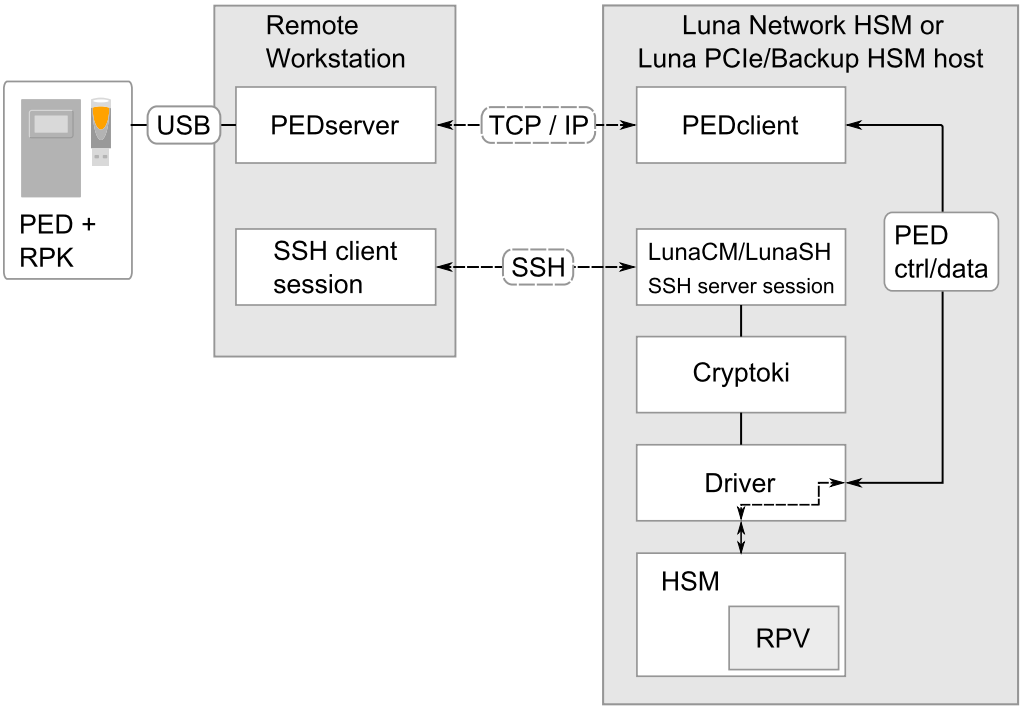
- SafeNet Luna Network HSM
- Host computer with SafeNet Luna PCIe HSM installed
- Host computer with USB-connected SafeNet Luna Backup HSM, configured for remote backup



Remote PED Connections

A SafeNet Luna PCIe HSM on a host computer running PEDclient can establish a Remote PED connection with any workstation that meets the following criteria:

- PEDServer is running
- a SafeNet Luna PED with firmware version 2.7.1 or newer is connected
- The orange PED key containing the Remote PED Vector (RPV) for that HSM is available



Priority and Lockout

If a Local PED connection is active and an operation is in progress, a Remote PED connection cannot be initiated until the active Local PED operation is completed. If the Local PED operation takes too long, the Remote PED command may time out.

When a Remote PED connection is active, the Local PED connection is ignored, and all authentication requests are routed to the Remote PED. Attempts to connect to a different Remote PED server are refused until the current connection times out or is deliberately ended. See "Ending or Switching the Remote PED Connection" on page 207.

One Connection at a Time

Remote PED can provide PED services to only one HSM at a time. To provide PED service to another HSM, you must first end the original Remote PED connection. See "Ending or Switching the Remote PED Connection" on page 207.

Timeout

PEDserver and PEDclient both have configurable timeout settings (default: 1800 seconds). See "pedserver mode config" on page 245 or "pedclient mode config" on page 229. The utilities are not aware of each other's timeout values, so the briefer value determines the actual timeout duration.

Once a partition has been Activated and cached the primary authentication (PED key) credential, the Crypto Officer or Crypto User can log in using only the secondary (alphanumeric) credentials and the Remote PED connection can be safely ended until the Partition SO needs to log in again.

Broken Connections

A Remote PED connection is broken if any of the following events occur:

> The connection is deliberately ended by the user

- The connection times out (default: 1800 seconds)
- SafeNet Luna PED is physically disconnected from its host
- VPN or network connection is disrupted
- You exit Remote PED mode on the Luna PED. If you attempt to change menus, the PED warns:

```
   ** WARNING **
Exiting now will
invalidate the RPK.
Confirm? YES/NO
```

If the link is broken, as long as the network connection is intact (or is resumed), you can restart PEDserver on the Remote PED host and run **ped connect** in LunaCM to re-establish the Remote PED link. In a stable network situation, the link will remain available until timeout.

PEDserver-PEDclient Communications

All communication between the Remote PED and the HSM is transmitted within an AES-256 encrypted channel, using session keys based on secrets shared out-of-band. This is considered a very secure query/response mechanism. The authentication conversation is between the HSM and the PED. Authentication data retrieved from the PED keys never exists unencrypted outside of the PED or the HSM. PEDclient and PEDserver provide the communication pathway between the PED and the HSM, and the data remains encrypted along that path.

Once the PED and HSM are communicating, they establish a common Data Encryption Key (DEK). DEK establishment is based on the Diffie-Hellman key establishment algorithm and a Remote PED Vector (RPV), shared between the HSM and the PED via the orange Remote PED Key (RPK). Once a common Diffie-Hellman value is established between the parties via the Diffie-Hellman handshake, the RPV is mixed into the value to create a 256-bit AES DEK on each side. If the PED and the HSM do not hold the same RPV, the resulting DEKs are different and communication is blocked.

Mutual authentication is achieved by exchanging random nonces, encrypted using the derived data encryption key. The authentication scheme operates as follows:

HSM	_	Remote PED
Send 8 bytes random nonce, R1, encrypted using the derived encryption key.	$\{R1 \parallel padding\}_{Ke}$ ->	
	<- $\{R2 \parallel R1\}_{Ke}$	Decrypt R1. Generate an 8 byte random nonce, R2. Concatenate R2 || R1 and encrypt the result using the derived encryption key.
Decrypt R2 || R1. Verify that received R1 value is the same as the originally generated value. Re-encrypt R2 and return it to Remote PED.	$\{padding \parallel R2\}_{Ke}$ ->	Verify that received R2 value is the same as the originally generated value.

Following successful authentication, the random nonce values are used to initialize the feedback buffers needed to support AES-OFB mode encryption of the two communications streams (one in each direction).

Sensitive data in transition between a PED and an HSM is end-to-end encrypted: plaintext security-relevant data is never exposed beyond the HSM and the PED boundaries at any time. The sensitive data is also hashed, using a SHA-256 digest, to protect its integrity during transmission.

Remote PED Setup

A Remote PED connection allows you to access PED-authenticated HSMs that are kept in a secure data center or other remote location where physical access is restricted or inconvenient. This section provides instructions for setting up different Remote PED configurations.

The procedure for setting up a Remote PED connection can be broken down into the following general steps:

1. "Initializing the Remote PED Vector (RPV) and Creating the Orange PED Key" below
2. "Installing PEDserver and Setting Up the Remote Luna PED" on the next page
3. "Opening a Remote PED Connection" on page 205
4. [OPTIONAL] "Ending or Switching the Remote PED Connection" on page 207

If you encounter issues with Remote PED, see "Remote PED Troubleshooting" on page 208.

Once Remote PED is set up, see "PED Key Management" on page 211.

Initializing the Remote PED Vector (RPV) and Creating the Orange PED Key

The Remote PED (via PEDserver) authenticates itself to the SafeNet Luna PCIe HSM with a randomly-generated encrypted value stored on an orange PED key. The orange key proves to the HSM that the Remote PED is authorized to perform authentication. A SafeNet Luna PCIe HSM administrator can create this key.

> **NOTE** Generally, the HSM SO creates an orange PED key (and backups), makes a copy for each valid Remote PED server, and distributes them to the Remote PED administrators.

If the HSM is already initialized, the HSM SO must log in to complete this procedure. You require:

- SafeNet Luna PED with firmware 2.7.1 or newer
- USB mini-B to USB-A connector cable
- Luna PED DC power supply (if included with your Luna PED)
- Blank or reusable orange PED key (or multiple keys, if you plan to make extra copies or use an M of N security scheme). See "Creating PED Keys" on page 212 for more information.

To initialize the RPV and create the orange PED key locally

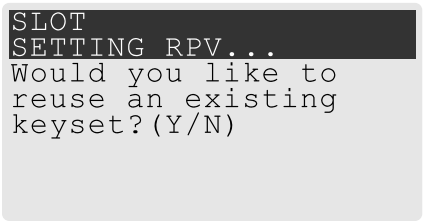
1. If you have not already done so, set up a Local PED connection (see "Local PED Setup" on page 196).
2. Launch LunaCM on the SafeNet Luna PCIe HSM host workstation.
3. If the HSM is initialized, login as HSM SO ("role login" on page 1). If not, skip to the next step.

   lunacm:>**role login -n so**
4. Ensure that you have the orange PED key(s) ready. Initialize the RPV ("ped vector" on page 1).

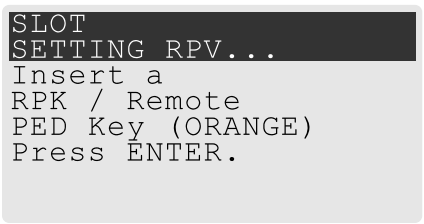
lunacm:> **ped vector init**

```
lunacm:>ped vector init
        You are about to initialize the Remote PED Vector
        Are you sure you wish to continue?

        Type 'proceed' to continue, or 'quit' to quit now ->proceed

        RPV was successfully initialized.

Command Result : No Error
```

5. Attend to the Luna PED and respond to the on-screen prompts. See "Creating PED Keys" on page 212 for a full description of the key-creation process.



- If you have an orange PED key with an existing RPV that you wish to use for this HSM, press **Yes**.
- If you are creating a new RPV, press **No**.



Continue following the prompts for PED PIN, M of N, and duplication options.

To continue setting up a Remote PED server, see "Installing PEDserver and Setting Up the Remote Luna PED" below.

Installing PEDserver and Setting Up the Remote Luna PED

The PEDserver software, installed on the Remote PED host workstation, allows the USB-connected Luna PED to communicate with remotely-located HSMs. The Remote PED administrator can install PEDserver using the Luna HSM Client installer. PEDserver is compatible with Windows operating systems only. You require:

> Network-connected workstation with compatible Windows operating system (refer to the Luna release 7.3 CRN)

> Luna HSM Client installer

> SafeNet Luna PED with firmware 2.7.1 or higher

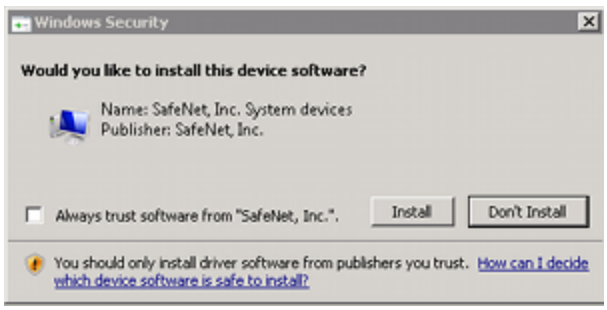
> USB mini-B to USB-A connector cable

> Luna PED DC power supply (PED 2.7.1 only; PED 2.8 and higher is powered by the USB connection)

To install PEDserver and the PED driver, and set up the Luna PED

1. Run the SafeNet Luna HSM Client installer and follow the on-screen instructions. When you reach the **Custom Setup** dialog box, select the **Luna Remote PED** option to be installed. Any additional installation choices are optional, for the purpose of this procedure.

2. When you are prompted to install the driver, click **Install**.



3. Reboot the computer to ensure that the Luna PED driver is accepted by Windows. This step is not required for Windows Server operating systems.
4. Connect the Luna PED to a USB port on the host system using the supplied USB mini-B to USB-A connector cable.

   PED version 2.8 and above is powered via the USB connection. If you are using PED version 2.7.1, connect it to power using the Luna PED DC power supply.

As soon as the PED receives power, it performs start-up and self-test routines (for PED v2.8 and later, the PED driver must be installed on the connected computer, or the display remains blank). It verifies the connection type and automatically switches to the appropriate operation mode when it receives the first command from the HSM.

To set the operation mode to **Remote PED** manually, see "Changing Modes" on page 195.

5. Open the Windows **Device Manager** to confirm that the Luna PED is recognized as **PED2**. If it appears as an unrecognized USB device:
   a. Disconnect the Luna PED from the host USB port.
   b. Reboot the computer to ensure that the Luna PED driver is accepted by Windows.
   c. Reconnect the Luna PED.

To continue setting up a Remote PED connection, see "Opening a Remote PED Connection" below.

Opening a Remote PED Connection

If you encounter issues, see "Remote PED Troubleshooting" on page 208.

The HSM/client administrator can use this procedure to establish an HSM-initiated Remote PED connection. You require:

- Administrative access to a network-connected workstation with PEDserver installed and Luna PED connected (see "Installing PEDserver and Setting Up the Remote Luna PED" on page 203)
- Administrative access to the SafeNet Luna PCIe HSM host via SSH
- Orange PED key with the HSM's RPV (see "Initializing the Remote PED Vector (RPV) and Creating the Orange PED Key" on page 202)

To open a Remote PED connection

1. Open an Administrator command prompt by right-clicking the Command Prompt icon and selecting **Run as administrator**. This step is not necessary if you are running Windows Server 20xx, as the Administrator prompt is launched by default.
2. Navigate to the SafeNet Luna HSM Client install directory.

   >**cd C:\Program Files\SafeNet\LunaClient**
3. Launch PEDserver (see "pedserver" on page 239 for all available options). If you are launching PEDserver on an IPv6 network, you must include the **-ip** option.

   >**pedserver mode start** [**-ip** <PEDserver_IP>]

   ```
   C:\Program Files\SafeNet\LunaClient>pedserver mode start
   Ped Server Version 1.0.6 (10006)
   Ped Server launched in startup mode.
   Starting background process
   Background process started
   Ped Server Process created, exiting this process.
   ```
4. Verify that the service has launched successfully ("pedserver mode" on page 244).

   >**pedserver mode show**

   Note the **Ped2 Connection Status**. If it says **Connected**, PEDserver is able to communicate with the Luna PED.

Note also the server port number (default: **1503**). You must specify this port along with the PEDserver host IP when you open a connection.

```
c:\Program Files\SafeNet\LunaClient>pedserver mode show
Ped Server Version 1.0.6 (10006)
Ped Server launched in status mode.

   Server Information:
      Hostname:                          DWG9999
      IP:                                0.0.0.0
      Firmware Version:                  2.7.1-5
      PedII Protocol Version:            1.0.1-0
      Software Version:                  1.0.6 (10006)

      Ped2 Connection Status:            Connected
      Ped2 RPK Count                     0
      Ped2 RPK Serial Numbers            (none)

   Client Information:                   Not Available

   Operating Information:
      Server Port:                       1503
      External Server Interface:         Yes
      Admin Port:                        1502
      External Admin Interface:          No

      Server Up Time:                    190 (secs)
      Server Idle Time:                  0 (secs) (0%)
      Idle Timeout Value:                1800 (secs)

      Current Connection Time:           0 (secs)
      Current Connection Idle Time:      0 (secs)
      Current Connection Total Idle Time: 0 (secs) (100%)
      Total Connection Time:             0 (secs)
      Total Connection Idle Time:        0 (secs) (100%)

Show command passed.
```

5. Use **ipconfig** to determine the PEDserver host IP. A static IP is recommended, but if you are connecting over a VPN, you may need to determine the current IP each time you connect to the VPN server.

   >**ipconfig**

6. Via SSH, launch LunaCM on the SafeNet Luna PCIe HSM host.
7. Initiate the Remote PED connection ("ped connect" on page 1).

   lunacm:>**ped connect -ip** <PEDserver_IP> **-port** <PEDserver_port> **-slot** <slot>

   > **NOTE** The **-slot** option may be required if you have multiple SafeNet Luna PCIe HSMs installed in one server. If you do not include this option, the currently-active slot is used.

   ```
   lunacm:>ped connect -ip 192.124.106.100 -port 1503

   Command Result : No Error
   ```

8. Issue the first command that requires authentication.
   - If the HSM is already initialized and you have the blue HSM SO key, log in ("role login" on page 1).

     lunacm:>**role login -name so**

- If the HSM is uninitialized, you can initialize it now ("hsm init" on page 1). Have blank or reusable blue and red PED keys ready (or multiple blue and red keys in case of M of N or if making multiple copies). See "Creating PED Keys" on page 212 for more information.

  lunacm:>**hsm init -label** <label>

9. The Luna PED prompts for an orange PED key. Present the orange PED key with the correct RPK.

```
SLOT
COMPUTE SESSION KEY.
Insert a
RPK / Remote
PED Key (ORANGE)
Press ENTER.
```

10. The Luna PED prompts for the key associated with the command you issued. Follow the on-screen directions to complete the authentication process.

```
SLOT
SO LOGIN...
Insert a
SO / HSM Admin
PED Key (BLUE)
Press ENTER.
```

> **NOTE** The Remote PED connection eventually times out (default: 1800 seconds), and must be re-initiated each time authentication is required. To simplify this process, you can set a default IP address and/or port for LunaCM to use each time you connect. To drop the Remote PED connection manually, see "Ending or Switching the Remote PED Connection" below.

11. [OPTIONAL] Set a default IP address and/or port for the SafeNet Luna PCIe HSM to look for a Remote PED host with PEDserver running ("ped set" on page 1).

    lunacm:>**ped set -ip** <PEDserver_IP> **-port** <PEDserver_port>

```
lunacm:>ped set -ip 192.124.106.100 -port 1503

Command Result : 0 (Success)
```

    With this default address set, the HSM administrator can use **ped connect** to initiate the Remote PED connection. The orange PED key may be required if the RPK has been invalidated since you last used it.

Ending or Switching the Remote PED Connection

PEDserver runs on the Remote PED host until explicitly stopped. PEDclient (running on the SafeNet Luna PCIe HSM host) has a default timeout period of 1800 seconds. If you want to connect to a different Remote PED server, or allow another HSM to use the current server, you must manually break the Remote PED connection.

To end or switch an HSM-initiated connection

1. End the Remote PED connection ("ped disconnect" on page 1).

   lunacm:>**ped disconnect**

```
lunacm:> ped disconnect
```

```
        Are you sure you wish to disconnect the remote ped?

        Type 'proceed' to continue, or 'quit' to quit now -> proceed

Command Result : No Error
```

2. You are now able to initiate a connection to a different Remote PED host running PEDserver ("ped connect" on page 1). You will need to present the orange PED key.

   lunacm:>**ped connect -ip** <PEDserver_IP> **-port** <port>

   > **NOTE** Running this command does not change the default Remote PED IP/port you may have previously set. If you want this new Remote PED server to be the default, set it using **ped set -ip** <PEDserver_IP> **-port** <port> (set "ped set" on page 1).

Remote PED Troubleshooting

If you encounter problems at any stage of the Remote PED connection process, refer to this section.

No Menu Appears on PED Display: Ensure Driver is Properly Installed

If the PED driver is not properly installed before connecting the PED to the workstation's USB port, the PED screen does not display the menu. If you encounter this problem, ensure that you have followed the entire procedure at "Installing PEDserver and Setting Up the Remote Luna PED" on page 203.

RC_SOCKET_ERROR: PEDserver Requires Administrator Privileges

If PEDserver is installed in the default Windows directory, it requires Administrator privileges to make changes. if you run PEDserver as an ordinary user, you may receive an error like the following:

```
c:\Program Files\SafeNet\LunaClient>pedserver mode start
Ped Server Version 1.0.6 (10006)
Ped Server launched in startup mode.
Starting background process
Failed to recv query response command: RC_SOCKET_ERROR c0000500
Background process failed to start : 0xc0000500 RC_SOCKET_ERROR
Startup failed. : 0xc0000500 RC_SOCKET_ERROR
```

To avoid this error, when opening a command line for PEDserver operations, right-click the Command Prompt icon and select **Run as Administrator**. Windows Server 20xx opens the Command Prompt as Administrator by default.

> **NOTE** If you do not have Administrator permissions on the Remote PED host, contact your IT department or install Luna HSM Client in a non-default directory (outside the **Program Files** directory) that is not subject to permission restrictions.

CKR_PED_UNPLUGGED: Reconnect Remote PED Before Issuing Commands

As described in the connection procedures, Remote PED connections time out after a default period of 1800 seconds (30 minutes). If you attempt PED authentication after timeout or after the connection has been broken for another reason, the Luna PED will not respond and you will receive an error like this:

```
lunacm:> role login -n so

        Please attend to the PED.
```

```
Error in execution: CKR_PED_UNPLUGGED.

Command Result : 0x8000002e (CKR_PED_UNPLUGGED)
```

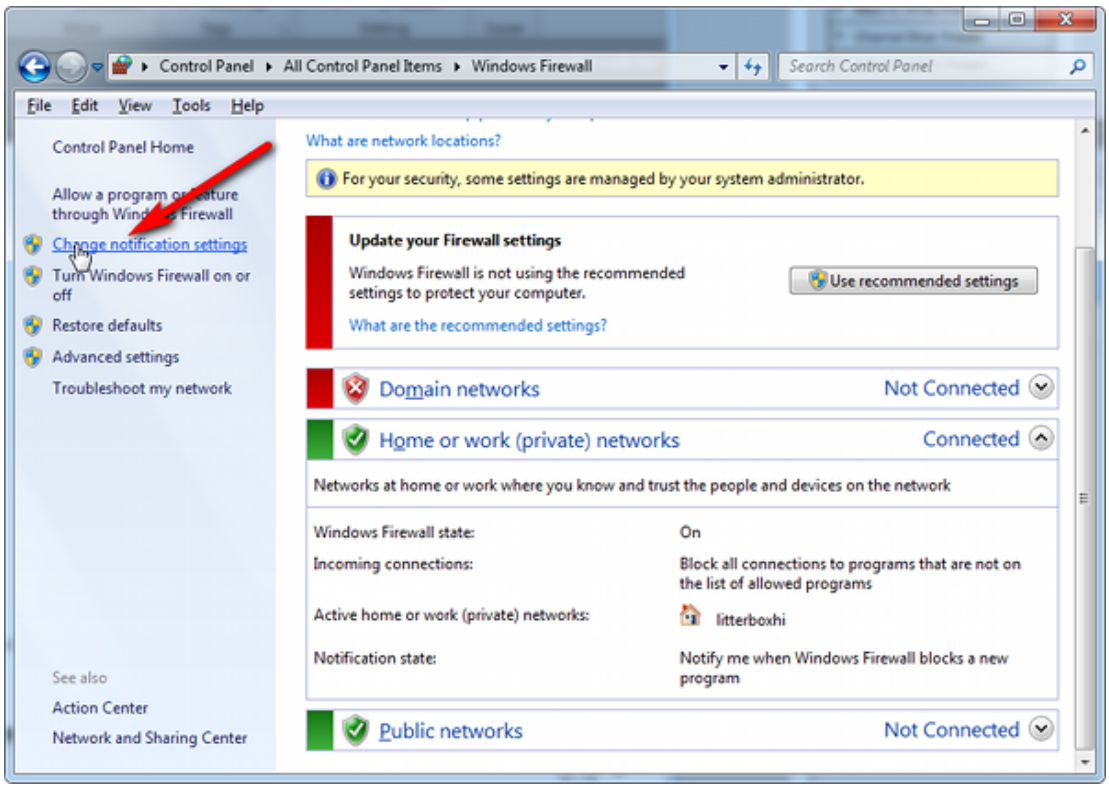
To avoid this error, re-initiate the connection before issuing any commands requiring PED authentication ("ped connect" on page 1):

lunacm:>**ped connect -ip** <PEDserver_IP> **-port** <PEDserver_port>

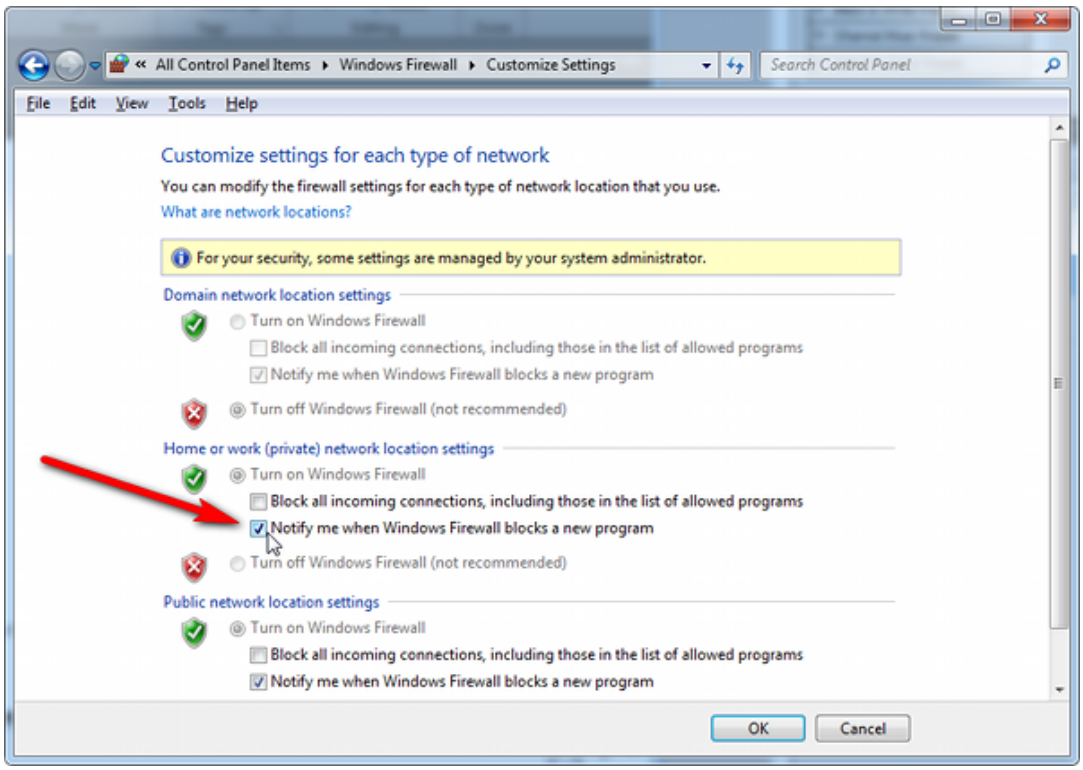
Remote PED Firewall Blocking

If you experience problems while attempting to configure a SafeNet Remote PED session over VPN, you might need to adjust Windows Firewall settings.

1. From the Windows Start Menu, select **Control Panel**.
2. Select **Windows Firewall**.
3. Select **Change notification settings**.



4. In the dialog **Customize settings for each type of network**, go to the appropriate section and activate **Notify me when Windows Firewall blocks a new program**.



With notifications turned on, a dialog box appears whenever Windows Firewall blocks a program, allowing you to override the block as Administrator. This allows PEDserver to successfully listen for PEDclient connections.

Remote PED Blocked Port Access

The network might be configured to block access to certain ports. If ports 1503 (the default PEDserver listening port) and 1502 (the administrative port) are blocked on your network, choose a different port when starting PEDserver, and when using **ped connect** (LunaCM) to initiate the Remote PED connection. Contact your network administrator for help.

You might choose to use a port-forwarding jump server, co-located with the SafeNet Luna PCIe HSM(s) on the datacenter side of the firewall. This can be a low-cost solution for port-blocking issues. It can also be used to implement a PKI authentication layer for Remote PED or other SSH access, by setting up smart-card access control to the jump server.

For example, you can use a standard Ubuntu Server distribution with OpenSSH installed and no other changes made to the standard installation with the following procedure:

1. Connect the Luna PED to a Windows host with SafeNet Luna HSM Client installed and PEDserver running.
2. Open an Administrator command prompt on the Remote PED host and start the port-forwarding service.

   >**plink -ssh -N -T -R 1600:localhost:1503 <user>@<Ubuntu_server_IP>**.
3. Launch LunaCM on the SafeNet Luna PCIe HSM host, and open the HSM-initiated connection ("ped connect" on page 1).

   lunacm:>**ped connect -ip <Ubuntu_server_IP> -port 1600**

The Remote PED host initiates the SSH session, via the Ubuntu jump server, which returns to the Remote PED host running PEDserver.

A variant of this arrangement also routes port 22 through the jump server, which allows administrative access to the SafeNet Luna PCIe HSM under the PKI access-control scheme.

ped connect Fails if IP is Not Accessible

On a system with two network connections, if PEDserver attempts to use an IP address that is not externally accessible, lunacm:>**ped connect** can fail. To resolve this:

1. Ensure that PEDserver is listening on the IP address that is accessible from outside.
2. If not, disable the network connection on which PEDserver is listening.
3. Restart PEDserver and confirm that it is listening on the IP address that is accessible from outside.

PEDserver on VPN fails

If PEDserver is running on a laptop that changes location, the active network address changes even though the laptop is not shutdown. If you unplugged from working at home, over the corporate VPN, commuted to the office, and reconnected the laptop there, PEDserver is still configured with the address you had while using the VPN. Running **pedserver -mode stop** does not completely clear all settings, so running **pedserver -mode start** again fails with a message like "Startup failed. : 0x0000303 RC_OPERATION_TIMED_OUT". To resolve this problem:

1. Close the current command prompt window.
2. Open a new Administrator command prompt.
3. Verify the current IP address.

   >**ipconfig**
4. Start PEDserver, specifying the new IP and port number ("pedserver mode start" on page 251).

   >**pedserver -mode start -ip** <new_IP> **-port** <port>

PED Key Management

Once you have established a Local or Remote PED connection, you can proceed with initializing roles on the HSM that require PED authentication. The procedures in this section will guide you through the PED prompts at each stage of PED key creation, PED authentication, and other operations with the SafeNet Luna PED.

> "Creating PED Keys" on the next page
  - "Stage 1: Reusing Existing PED Keys" on page 213
  - "Stage 2: Defining M of N" on page 214
  - "Stage 3: Setting a PED PIN" on page 215
  - "Stage 4: Duplicating New PED Keys" on page 216
> "Performing PED Authentication" on page 217
> "Consequences of Losing PED Keys" on page 218
> "Identifying a PED Key Secret" on page 221
> "Duplicating Existing PED Keys" on page 222
> "Changing a PED Key Secret" on page 222

Creating PED Keys

When you initialize an HSM, partition, or role, the SafeNet Luna PED issues a series of prompts for you to follow to create your PED keys. PED key actions have a timeout setting (default: 200 seconds); ensure that you have everything you need before issuing an initialization command. The requirements for the operation depend on the PED key scheme you have chosen in advance, based on your organization's security policy. Consider these guidelines before you begin:

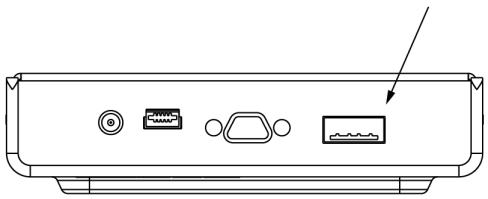
> If you are reusing an existing PED key or keyset, the owners of those keys must be present with their keys and PED PINs ready.

> If you plan to use an M of N authentication scheme (quorum, or split-secret), all the parties involved must be present and ready to create their authentication split. It is advisable for each key holder to create backup duplicates, so you must have a sufficient number of blank or rewritable PED keys ready before you begin.

> If you plan to make backup duplicates of PED keys, you must have a sufficient number of blank or rewritable PED keys ready.

> If you plan to use PED PINs, ensure that they can be privately entered on the Luna PED and memorized, or written down and securely stored.

Whenever the Luna PED prompts you to insert a PED key, use the USB port on the top of the PED:



To initiate PED key creation

1. Issue one of the following LunaCM commands to initialize the applicable role, domain, or vector.
   - **Blue HSM SO and Red HSM Domain Keys** ("hsm init" on page 1):

     lunacm:>**hsm init**
   - **Orange Remote PED Key** ("ped vector" on page 1):

     lunacm:>**ped vector init**
   - **Blue Partition SO and Red Partition Domain Keys** ("partition init" on page 1):

     lunacm:>**partition init**
   - **Black Crypto Officer Key** ("role init" on page 1):

     lunacm:>**role init -name co**
   - **Gray Crypto User Key** ("role init" on page 1):

     lunacm:>**role init -name cu**
   - **White Audit User Key** ("role init" on page 1):

     lunacm:>**role init -name au**

   The Luna PED responds, displaying:

```
Remote PED mode
Token found
```

2. Follow the PED prompts in the following four stages.

Stage 1: Reusing Existing PED Keys

If you want to use a PED key with an existing authentication secret, have the key ready to present to the PED. Reasons for reusing keys may include:

> You want to use the same blue SO key to authenticate multiple HSMs/partitions

> You want to initialize a partition in an already-existing cloning domain (to be part of an HA group)

> **CAUTION!** The initialization procedure is the only opportunity to set the HSM/partition's cloning domain. It cannot be changed later without reinitializing the HSM, or deleting and recreating the partition. Ensure that you have the correct red key(s) ready.

See "Shared PED Key Secrets" on page 191 and "Domain PED Keys" on page 191 for more information.

1. The first PED prompt asks if you want to reuse an existing PED key. Press **Yes** or **No** on the keypad to continue.

```
SLOT
SETTING SO PIN...
Would you like to
reuse an existing
keyset?(Y/N)
```

- If you select **No**, skip to "Stage 2: Defining M of N" on the next page.
- If you select **Yes**, the PED prompts you for a key. Insert the key you want to reuse and press **Enter**.

```
SLOT
SETTING SO PIN...
Insert a
SO / HSM Admin
PED Key (BLUE)
Press ENTER.
```

2. If the key has a PED PIN, the PED prompts you to enter it now. Enter the PIN on the keypad and press **Enter**.

```
SLOT
READING SO PIN...
Enter PED PIN:
*********
```

3. If the key is part of an M of N scheme, the PED prompts you for the next key. You must present enough key splits (M) to reconstitute the entire authentication secret.

```
SLOT
READING SO PIN...
Keys read: 01 of 03
Insert another
SO / HSM Admin
PED Key (BLUE)
Press ENTER.
```

4. The PED asks if you want to create a duplicate set of keys. If you are duplicating an M of N keyset, you need a number of blank or rewritable keys equal to N.

```
SLOT
READING SO PIN...
Are you duplicating
this keyset?(Y/N)
Warning: You will
need all N keys!
```

- If you select **No**, the process is complete.
- If you select **Yes**, complete "Stage 3: Setting a PED PIN" on the next page for all the duplicate keys you want.

Stage 2: Defining M of N

If you chose to create a new keyset, the Luna PED prompts you to define the M of N scheme (quorum and pool of splits) for the role, domain, or vector. See "M of N Split Secrets (Quorum)" on page 192 for more information. If you do not want to use M of N (authentication by one PED key), enter a value of **1** for both M and N.

1. The PED prompts you to enter a value for M (the minimum number of split-secret keys required to authenticate the role, domain, or vector - the quorum). Set a value for M by entering it on the keypad and pressing **Enter**. If you are not using an M of N scheme, enter "**1**".

```
SLOT
SETTING SO PIN...
M value? (1-16)

>03
```

2. The PED prompts you to enter a value for N -- the total number of split-secret keys you want to create (the pool of splits from which a quorum will be drawn). Set a value for N by entering it on the keypad and pressing **Enter**. If you are not using an M of N scheme, enter "**1**".

```
SLOT
SETTING SO PIN...
N value? (M-16)

>05
```

3. Continue to "Stage 3: Setting a PED PIN" below. You must complete stage 3 for each key in the M of N scheme.

Stage 3: Setting a PED PIN

If you are creating a new key or M of N split, you have the option of setting a PED PIN that must be entered by the key owner during authentication. PED PINs must be 4-48 digits long. Do not use 0 for the first digit. See "PED PINs" on page 192 for more information.

> **CAUTION!** If you forget your PED PIN, it is the same as losing the PED key entirely; you cannot authenticate the role. See "Consequences of Losing PED Keys" on page 218.

1. The PED prompts you to insert a blank or reusable PED key. If you are creating an M of N split, the number of already-created splits is displayed.

```
SLOT
SETTING SO PIN...
Insert a
SO / HSM Admin
PED Key (BLUE)
Press ENTER.
```

```
SLOT
SETTING SO PIN...
Keys write: 03 of 05
Insert another
SO / HSM Admin
PED Key (BLUE)
Press ENTER.
```

2. Insert the PED key and press **Enter**. The PED prompts for confirmation.

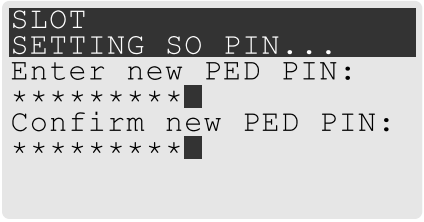
```
SLOT
SETTING SO PIN...
   ** WARNING **
This PED Key is
blank.
Overwrite? YES/NO
```

If the PED key you inserted is not blank, you must confirm twice that you want to overwrite it.

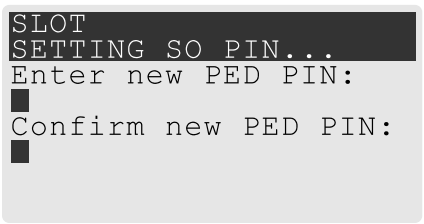
```
SLOT
SETTING SO PIN...
   ** WARNING **
This PED Key is for
Domain.
Overwrite? YES/NO
```

```
SLOT
SETTING SO PIN...
   ** WARNING **
Are you sure you
want to overwrite
this PED key? YES/NO
```

3. The PED prompts you for a PIN.
   - If you want to set a PED PIN, enter it on the keypad and press **Enter**. Enter the PIN again to confirm it.



- If you do not want to set a PED PIN, press **Enter** twice without entering anything on the keypad. You will not be asked to enter a PIN for this key in the future.



4. If there are more keys in the M of N scheme, repeat this stage. Otherwise, continue to "Stage 4: Duplicating New PED Keys" below.

Stage 4: Duplicating New PED Keys

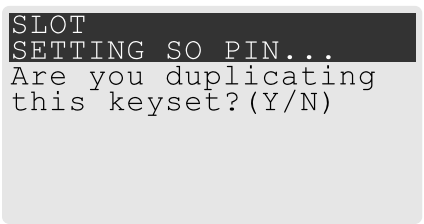
You now have the option to create duplicates of your newly-created PED key(s). There are two reasons to do this now:

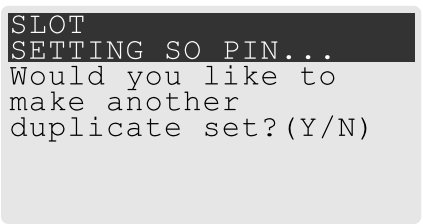
> If you want more than one person to be able to authenticate a role, you can create multiple keys for that role now, with each person being able to set their own PED PIN. Duplicates you create later are intended as backups, and will have the same PED PIN (or none) as the key they are copied from.

> In case of key loss or theft.

You can make backups now or later. See also "Duplicating Existing PED Keys" on page 222.

1. The next PED prompt asks if you want to create a duplicate keyset (or another duplicate). Press **Yes** or **No** on the keypad to continue.





- If you select **No**, the key creation process is complete.
- If you select **Yes**, complete "Stage 3: Setting a PED PIN" on the previous page for the duplicate keyset. You can set the same PED PIN to create a true copy, or set a different PED PIN for each duplicate.

2. If you specified an M of N scheme, you are prompted to repeat "Stage 3: Setting a PED PIN" on the previous page for each M of N split. Otherwise, the key creation process is complete.

Performing PED Authentication

When connected, the SafeNet Luna PED responds to authentication commands in LunaCM. Commands that require PED actions include:

- Role login commands (blue, black, gray, or white PED keys)
- Backup/restore commands (red PED keys)
- Remote PED connection commands (orange PED key)

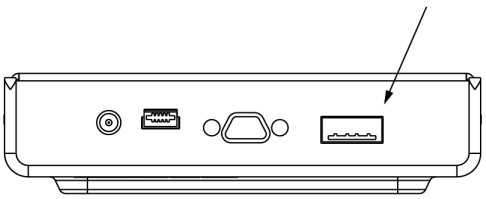
When you issue a command that requires PED interaction, the interface returns a message like the following:

```
lunacm:>role login -name po

        Please attend to the PED.
```

The PED briefly displays the following message before prompting you for the appropriate PED key:

```
Remote PED mode
Token found
```

Whenever the Luna PED prompts you to insert a PED key, use the USB port on the top of the PED:



> **CAUTION!** Multiple failed authentication attempts result in zeroization of the HSM or partition, or role lockout, depending on the role. This is a security measure designed to thwart repeated, unauthorized attempts to access cryptographic material. For details, see "Failed Login Attempts" on page 329.

To perform PED authentication

1. The PED prompts for the corresponding PED key. Insert the PED key (or the first M of N split-secret key) and press **Enter**.

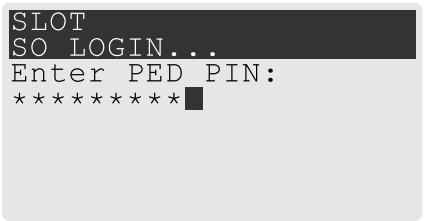
```
lunacm:>role login -name po

        Please attend to the PED.
```

```
SLOT
SO LOGIN...
Insert a
SO / HSM Admin
PED Key (BLUE)
Press ENTER.
```

- If the key you inserted has an associated PED PIN, continue to step 2.
- If the key you inserted has no PED PIN, but it is an M of N split, skip to step 3.
- Otherwise, authentication is complete and the PED returns control to the command interface.

```
Command Result : No Error
```

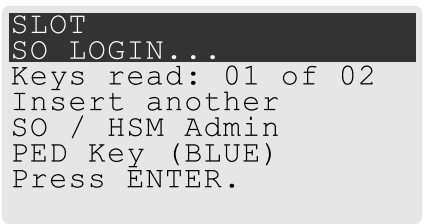
2. The PED prompts for the PED PIN. Enter the PIN on the keypad and press **Enter**.



- If the key you inserted is an M of N split, continue to step 3.
- Otherwise, authentication is complete and the PED returns control to the command interface.

```
Command Result : No Error
```

3. The PED prompts for the next M of N split-secret key. Insert the next PED key and press **Enter**.



- If the key you inserted has an associated PED PIN, return to step 2.
- Repeat steps 2 and/or 3 until the requisite M number of keys have been presented to the PED. At this point, authentication is complete and the PED returns control to the command interface.

```
Command Result : No Error

lunacm:>
```

Consequences of Losing PED Keys

PED keys are the only means of authenticating roles, domains, and RPVs on the PED-authenticated SafeNet Luna PCIe HSM. Losing a PED keyset effectively locks the user out of that role. Always keep secure backups of your PED keys, including M of N split secrets. Forgetting the PED PIN associated with a key is equivalent to losing the key entirely. Losing a split-secret key is less serious, unless enough splits are lost so that M cannot be satisfied.

If a PED key is lost or stolen, log in with one of your backup keys and change the existing PED secret immediately, to prevent unauthorized HSM access.

The consequences of a lost PED key with no backup vary depending on the type of secret:

> "Blue HSM SO Key" on the next page

> "Red HSM Domain Key" on the next page

> "Orange Remote PED Key" below
> "Blue Partition SO Key" below
> "Red Partition Domain Key" on the next page
> "Black Crypto Officer Key" on the next page
> "Gray Crypto User Key" on page 221
> "White Audit User Key" on page 221

Blue HSM SO Key

If the HSM SO secret is lost, you can no longer perform administrative tasks on the HSM, including partition creation and client assignment. If you use the same blue SO key for your HSM backup partitions, the contents of the HSM Admin partition are unrecoverable. Take the following steps:

1. Contact all Crypto Officers and have them immediately make backups of their existing partitions.
2. When all important partitions are backed up, execute a factory reset of the HSM.
3. Initialize the HSM and create a new HSM SO secret. Use the original red HSM cloning domain key.
4. Restore the HSM Admin partition contents from a recent backup, if you have one.
5. Recreate the partitions and reassign them to their respective clients.
6. Partition SOs must initialize the new partitions using their original blue and red key(s), and initialize the Crypto Officer role (and Activation secret, if applicable). Supply the new black CO keys to the Crypto Officers.
7. Crypto Officers must change the login credentials from the new black CO key to their original black keys (and reset the Activation secret password, if applicable).
8. Crypto Officers can now restore all partition contents from backup.
9. If you are using Remote PED, you must recreate the Remote PED Vector (RPV). Reuse the original orange key.

Red HSM Domain Key

If the HSM Key Cloning Vector is lost, you can no longer perform backup/restore operations on the HSM Admin partition(s). If the HSM is factory-reset, the contents of the HSM Admin partition are unrecoverable. Follow the same procedure as you would if you lost the blue HSM SO key, but you cannot restore the HSM Admin partition from backup.

Orange Remote PED Key

If the Remote PED Vector is lost, create a new one and distribute a copy to the administrator of each Remote PED server. See "Initializing the Remote PED Vector (RPV) and Creating the Orange PED Key" on page 202.

Blue Partition SO Key

If the Partition SO secret is lost, you can no longer perform administrative tasks on the partition. Take the following steps:

1. Have the Crypto Officer immediately make a backup of the partition objects.
2. Have the HSM SO delete the partition, create a new one, and assign it to the same client.

3. Initialize the new partition with a new blue Partition SO key and the original red cloning domain key(s).
4. Initialize the Crypto Officer role (and Activation secret, if applicable). Supply the new black CO key to the Crypto Officer.
5. The Crypto Officer must change the login credentials from the new black CO key to their original black key (and reset the Activation secret password, if applicable).
6. The Crypto Officer can now restore all partition contents from backup.

Red Partition Domain Key

If the Partition Key Cloning Vector is lost, you can no longer perform backup/restore operations on the partition (s), or make changes to HA groups in that cloning domain. You can still perform all other operations on the partition. Take the following steps:

1. Have the HSM SO create a new partition (or multiple partitions, to replace the entire HA group) and assign it to the same client(s).
2. Initialize the partition(s)with a new cloning domain.
3. Initialize the Crypto Officer role with the original black Crypto Officer key (and Activation password, if applicable).
4. Create objects on the new partition to replace those on the original partition.
5. As soon as possible, change all applications to use the objects on the new partition.
6. When objects on the original partition are no longer in production use, the HSM SO can delete the original partition.

Black Crypto Officer Key

If the Crypto Officer secret is lost, you can no longer create objects on the partition, or perform backup/restore operations. You might still be able to use the partition, depending on the following criteria:

> **PIN reset by Partition SO:**

- If HSM policy **15: Enable SO reset of partition PIN** is set to **1**, the Partition SO can reset the Crypto Officer secret and create a new black CO key.

  lunacm:>**role resetpw -name co**

- If this policy is set to **0** (default), the CO is locked out unless other criteria in this list apply.

> **Partition Activation:**

- If the partition is Activated, you can still access it for production using the CO challenge secret. Change your applications to use objects on a new partition as soon as possible.
- If the partition is not Activated, read-only access of essential objects might still be available via the Crypto User role.

> **Crypto User**

- If the Crypto User is initialized, you can use the CU role for read-only access to essential partition objects while you change your applications to use objects on a new partition.

If none of these criteria apply, the contents of the partition are unrecoverable.

Gray Crypto User Key

If the Crypto User secret is lost, the Crypto Officer can reset the CU secret and create a new gray key:

lunacm:>**role resetpw -name cu**

White Audit User Key

If the Audit User secret is lost, you can no longer cryptographically verify existing audit logs or make changes to the audit configuration. The existing logs can still be viewed. Re-initialize the Audit User role on the affected HSMs, using the same white key for HSMs that will verify each other's logs.

Identifying a PED Key Secret

You can use this procedure to identify the type of secret (role, domain, or RPV) stored on an unidentified PED key. This procedure will not tell you:

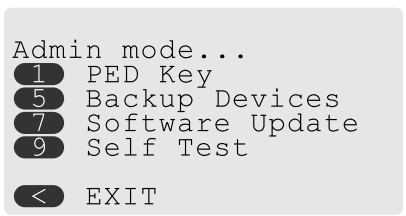
- identifying information about the HSM the key is associated with
- whether the key is part of an M of N scheme, or how many keys are in the set
- whether the key has a PED PIN assigned
- who the key belongs to

You require:

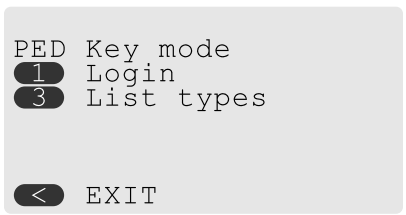
- SafeNet Luna PED in Admin Mode (see "Changing Modes" on page 195)
- the key you want to identify

To identify the type of secret stored on a PED key

1. Insert the PED key you want to identify.
2. From the Admin mode menu, press **1** on the keypad to select the **PED Key** option.



3. From the PED Key mode menu, press **3** on the keypad to select the **List types** option.



The PED secret type is identified on-screen.

```
PED Key mode
Found keys:
Domain


Press ENTER.
```

Duplicating Existing PED Keys

During the key creation process, you have the option to create multiple copies of PED keys. If you want to make backups of your keys later, you can use this procedure to copy PED keys. You require:

- SafeNet Luna PED in Admin Mode (see "Changing Modes" on page 195)
- Enough blank or rewritable keys to make your copies

The PED key is duplicated exactly by this process. If there is a PED PIN assigned, the same PIN is assigned to the duplicate key. If the key is part of an M of N scheme, the duplicates may not be used in the same login process to satisfy the M of N requirements. You must also have copies of the other keys in the M of N keyset. See "M of N Split Secrets (Quorum)" on page 192.

To duplicate an existing PED key

1. Insert the PED key you want to duplicate. Have a blank or rewritable PED key ready.
2. From the Admin mode menu, press **1** on the keypad to login to the PED key.

```
PED Key mode
1 Login
3 List types


< EXIT
```

3. Press **7** on the keypad and follow the on-screen instructions.

```
PED Key mode
          2 Logout
3 List types

7 Duplicate

< EXIT
```

Changing a PED Key Secret

It may be necessary to change the PED secret associated with a role. Reasons for changing credentials include:

- Regular credential rotation as part of your organization's security policy
- Compromise of a role due to loss or theft of a PED key
- Personnel changes in your organization or changes to individual security clearances
- Changes to your security scheme (implementing/revoking M of N, PED PINs, or shared secrets)

The procedure for changing a PED key credential depends on the type of key. Procedures for each type are provided below.

> **CAUTION!** If you are changing a PED credential that is shared among multiple HSMs/partitions/roles, always keep at least one copy of the old keyset until the affected HSMs/partitions/roles are all changed to the new credential. When changing PED credentials, you must always present the old keyset first; do not overwrite your old PED keys until you have no further need for them.

> "Blue HSM SO Key" below
> "Red HSM Domain Key" below
> "Orange Remote PED Key" below
> "Blue Partition SO Key" on the next page
> "Red Partition Domain Key" on the next page
> "Black Crypto Officer Key" on the next page
> "Gray Crypto User Key" on the next page
> "White Audit User Key" on page 225

Blue HSM SO Key

The HSM SO can use this procedure to change the HSM SO credential.

To change the blue HSM SO PED key credential

1. In LunaCM, set the active slot to the Admin partition and login as HSM SO ("role login" on page 1).

   lunacm:>**role login -name so**

2. Initiate the PED key change ("role changepw" on page 1).

   lunacm:>**role changepw -name so**

3. You are prompted to present the original blue key(s) and then to create a new HSM SO keyset. See "Creating PED Keys" on page 212.

Red HSM Domain Key

It is not possible to change an HSM's cloning domain without factory-resetting the HSM and setting the new cloning domain as part of the standard initialization procedure.

> **CAUTION!** If you set a different cloning domain for the HSM, you cannot restore the HSM Admin partition from backup.

Orange Remote PED Key

The HSM SO can use this procedure to change the Remote PED Vector (RPV) for the HSM.

To change the RPV/orange key credential

1. In LunaCM, set the active slot to the Admin partition and login as HSM SO ("role login" on page 1).

lunacm:>**role login -name so**

2. Initialize the RPV ("ped vector" on page 1).

   lunacm:>**ped vector init**

   You are prompted to create a new Remote PED key.

3. Distribute a copy of the new orange key to the administrator of each Remote PED server.

Blue Partition SO Key

The Partition SO can use this procedure to change the Partition SO credential.

To change a blue Partition SO PED key credential

1. In LunaCM, log in as Partition SO ("role login" on page 1).

   lunacm:>**role login -name po**

2. Initiate the PED key change ("role changepw" on page 1).

   lunacm:>**role changepw -name po**

3. You are prompted to present the original blue key(s) and then to create a new Partition SO keyset.

Red Partition Domain Key

It is not possible to change a partition's cloning domain. A new partition must be created and initialized with the desired domain. The new partition will not have access to any of the original partition's backups. It cannot be made a member of the same HA group as the original.

Black Crypto Officer Key

The Crypto Officer can use this procedure to change the Crypto Officer credential.

To change a black Crypto Officer PED key credential

1. In LunaCM, log in as Crypto Officer ("role login" on page 1).

   lunacm:>**role login -name co**

2. Initiate the PED key change ("role changepw" on page 1).

   lunacm:>**role changepw -name co**

3. You are prompted to present the original black key(s) and then to create a new Crypto Officer keyset.

Gray Crypto User Key

The Crypto User can use this procedure to change the Crypto User credential.

To change a gray Crypto User PED key credential

1. In LunaCM, log in as Crypto User ("role login" on page 1).

   lunacm:>**role login -name cu**

2. Initiate the PED key change ("role changepw" on page 1).

   lunacm:>**role changepw -name cu**

3. You are prompted to present the original gray key(s) and then to create a new Crypto User keyset.

White Audit User Key
The Audit User can use this procedure to change the Audit User credential.

To change the white Audit User PED key credential

1. In LunaCM, set the active slot to the Admin partition and login as Auditor ("role login" on page 1).

   lunacm:>**role login -name au**

2. Initiate the PED key change ("role changepw" on page 1).

   lunacm:>**role changepw -name au**

3. You are prompted to present the original white key(s) and then to create a new Audit User keyset.

PEDserver and PEDclient

You can use the **PEDserver** and **PEDclient** utilities to manage your remote PED devices.

The PEDserver Utility

PEDserver is required to run on any computer that has a SafeNet Remote PED attached, and is providing PED services.

The PEDserver utility has one function. It resides on a computer with an attached Luna PED (in Remote Mode), and it serves PED operations to an instance of PEDclient that operates on behalf of an HSM. The HSM could be local to the computer that has PEDserver running, or it could be on another HSM host computer at some distant location.

PEDserver can also run in peer-to-peer mode, where the server initiates the connection to the Client. This is needed when the Client (usually SafeNet Luna Network HSM) is behind a firewall that forbids outgoing initiation of connections. See "Remote Backup Service" on page 51 for more information.

See "pedserver" on page 239.

The PEDclient Utility

PEDclient is required to run on any host of an HSM that needs to be served by a Remote Luna PED. PEDclient must also run on any host of a Remote Backup HSM that will be serving remote primary HSMs.

The PEDclient utility performs the following functions:

- It mediates between the HSM where it is installed and the Luna PED where PEDserver is installed, to provide PED services to the requesting HSM(s).
- It resides on a computer with RBS and an attached SafeNet Luna Backup HSM, and it connects with another instance of PEDclient on a distant host of an HSM, to provide the link component for Remote Backup Service.
- It acts as the logging daemon for HSM audit logs.

Thus, for example, in the case where an administrative workstation or laptop has both a Remote PED and a Remote Backup HSM attached, PEDclient would perform double duty. It would link with a locally-running instance of PEDserver, to convey HSM requests from the locally-connected Backup HSM to the locally-connected PED, and return the PED responses. As well, it would link a locally-running instance of RBS and a distant PEDclient instance to mediate Remote Backup function for that distant HSM's partitions. See "Remote Backup Service" on page 51 in the *Administration Guide* for more information.

See "pedclient" below.

pedclient

Use the **pedclient** commands to start, stop, and configure the PEDclient service.

Syntax

pedclient mode

- **assignid**
- **config**
- **deleteid**
- **releaseid**
- **setid**
- **show**
- **start**
- **stop**
- **testid**

Option	Description
assignid	Assigns a PED ID mapping to an HSM. See "pedclient mode assignid" on page 228.
config	Modifies or shows existing configuration file settings. See "pedclient mode config" on page 229.
deleteid	Deletes a PED ID mapping. See "pedclient mode deleteid" on page 231.
releaseid	Releases a PED ID mapping from an HSM. See "pedclient mode releaseid" on page 232.
setid	Creates a PED ID mapping. See "pedclient mode setid" on page 233.
show	Queries if PEDclient is currently running and gets details about PEDclient. See "pedclient mode show" on page 234.
start	Starts up PEDclient. See "pedclient mode start" on page 235.
stop	Shuts down PEDclient. See "pedclient mode stop" on page 237.

Option	Description
testid	Tests a PED ID mapping. See "pedclient mode testid" on page 238.

pedclient mode assignid

Assigns a PED ID mapping to a specified HSM.

Syntax

pedclient mode assignid -id <pedid> **-id_serialnumber** <serial> [**-logfilename** <filename>] [**-loginfo** <0 or 1>] [**-logwarning** <0 or 1>] [**-logerror** <0 or 1>] [**-logtrace** <0 or 1>] [**-maxlogfilesize** <size>] [**-locallogger**]

Option	Description
-id <pedid>	Specifies the ID of the PED to be assigned.
-id_serialnumber <serial>	Specifies the serial number of the HSM to be linked to the specified PED ID.
-logfilename <filename>	Optional. Specifies the log file name to which the logger should log messages.
-loginfo <0 or 1>	Optional. Specifies if the logger should log "info" messages. Set to 0 for no, 1 for yes.
-logwarning <0 or 1>	Optional. Specifies if the logger should log "warning" messages. Set to 0 for no, 1 for yes.
-logerror <0 or 1>	Optional. Specifies if the logger should log "error" messages. Set to 0 for no, 1 for yes.
-logtrace <0 or 1>	Optional. Specifies if the logger should log "trace" messages. Set to 0 for no, 1 for yes.
-maxlogfilesize <size>	Optional. Specifies the maximum log file size in KB.
-locallogger	Optional. Specifies that the Remote PED logger should be used, not the IS logging system.

Example

```
C:\Program Files\Safenet\LunaClient>pedClient –mode assignid -id 1234 -id_serialnumber 123456789
```

pedclient mode config

Modifies or shows existing configuration file settings.

Syntax

pedclient mode config -show -set [**-eadmin** <0 or 1>] [**-idletimeout** <int>] **[-ignoreideltimeout]** [**-socketreadtimeout** <int>] [**-socketwritetimeout** <int>] [**-shutdowntimeout** <int>] [**-pstartuptimeout** <int>] [**-pshutdowntimeout** <int>] [**-logfilename** <filename>] [**-loginfo** <0 or 1>] [**-logwarning** <0 or 1>] [**-logtrace** <0 or 1>] [**-maxlogfilesize** <size>] [**-locallogger**]

Option	Description
-show	Displays the contents of the configuration file.
-set	Updates the configuration file to be up to date with other supplied options.
-eadmin <0 or 1>	Optional. Specifies if the administration port is on "localhost" or on the external host name.
-idletimeout <int>	Optional. Specifies the idle connection timeout, in seconds.
-ignoreidletimeout	Optional. Specifies that the idle connection timeout should not apply to the connection established between the PED and HSM during their assignment.
-socketreadtimeout <int>	Optional. Specifies the socket read timeout, in seconds.
-socketwritetimeout <int>	Optional. Specifies the socket write timeout, in seconds.
-shutdowntimeout <int>	Optional. Specifies the shutdown timeout for internal services, in seconds.
-pstartuptimeout <int>	Optional. Specifies the startup timeout for the detached process, in seconds.
-pshutdowntimeout <int>	Optional. Specifies the shutdown timeout for the detached process, in seconds.
-logfilename <filename>	Optional. Specifies the log file name to which the logger should log messages.
-loginfo <0 or 1>	Optional. Specifies if the logger should log "info" messages. Set to 0 for no, 1 for yes.
-logwarning <0 or 1>	Optional. Specifies if the logger should log "warning" messages. Set to 0 for no, 1 for yes.
-logerror <0 or 1>	Optional. Specifies if the logger should log "error" messages. Set to 0 for no, 1 for yes.
-logtrace <0 or 1>	Optional. Specifies if the logger should log "trace" messages. Set to 0 for no, 1 for yes.

Option	Description
-maxlogfilesize <size>	Optional. Specifies the maximum log file size in KB.
-locallogger	Optional. Specifies that the Remote PED logger should be used, not the IS logging system.

Example

```
C:\Program Files\Safenet\LunaClient>pedClient –mode config -show
```

pedclient mode deleteid

Deletes a PED ID mapping between a specified PED and PEDserver.

Syntax

pedclient mode deleteid -id <PED_ID> [**-logfilename** <filename>] [**-loginfo** <0 or 1>] [**-logwarning** <0 or 1>] [**-logerror** <0 or 1>] [**-logtrace** <0 or 1>] [**-maxlogfilesize** <size>] [**-locallogger**]

Option	Description
-id <PED_ID>	Specifies the ID of the PED to be deleted from the map.
-logfilename <filename>	Optional. Specifies the log file name to which the logger should log messages.
-loginfo <0 or 1>	Optional. Specifies if the logger should log "info" messages. Set to 0 for no, 1 for yes.
-logwarning <0 or 1>	Optional. Specifies if the logger should log "warning" messages. Set to 0 for no, 1 for yes.
-logerror <0 or 1>	Optional. Specifies if the logger should log "error" messages. Set to 0 for no, 1 for yes.
-logtrace <0 or 1>	Optional. Specifies if the logger should log "trace" messages. Set to 0 for no, 1 for yes.
-maxlogfilesize <size>	Optional. Specifies the maximum log file size in KB.
-locallogger	Optional. Specifies that the Remote PED logger should be used, not the IS logging system.

Example

```
C:\Program Files\Safenet\LunaClient>pedClient –mode deleteid -id 1234
```

pedclient mode releaseid

Releases a PED ID mapping from the HSM it was assigned to.

Syntax

pedclient mode releaseid -id <PED_ID> [**-logfilename** <filename>] [**-loginfo** <0 or 1>] [**-logwarning** <0 or 1>] [**-logerror** <0 or 1>] [**-logtrace** <0 or 1>] [**-maxlogfilesize** <size>] [**-locallogger**]

Option	Description
-id <PED_ID>	Specifies the ID of the PED to be released.
-logfilename <filename>	Optional. Specifies the log file name to which the logger should log messages.
-loginfo <0 or 1>	Optional. Specifies if the logger should log "info" messages. Set to 0 for no, 1 for yes.
-logwarning <0 or 1>	Optional. Specifies if the logger should log "warning" messages. Set to 0 for no, 1 for yes.
-logerror <0 or 1>	Optional. Specifies if the logger should log "error" messages. Set to 0 for no, 1 for yes.
-logtrace <0 or 1>	Optional. Specifies if the logger should log "trace" messages. Set to 0 for no, 1 for yes.
-maxlogfilesize <size>	Optional. Specifies the maximum log file size in KB.
-locallogger	Optional. Specifies that the Remote PED logger should be used, not the IS logging system.

Example

```
C:\Program Files\Safenet\LunaClient>pedClient –mode releaseid -id 1234
```

pedclient mode setid

Creates a PED ID mapping between a specified PED and PEDserver.

Syntax

pedclient mode setid -id <PED_ID> **-id_ip** <hostname> **-id_port** <port> [**-logfilename** <filename>] [**-loginfo** <0 or 1>] [**-logwarning** <0 or 1>] [**-logerror** <0 or 1>] [**-logtrace** <0 or 1>] [**-maxlogfilesize** <size>] [**-locallogger**]

Option	Description
-id <PED_ID>	Specifies the ID of the PED to be mapped.
-id_ip <hostname>	Specifies the IP address or hostname of the PED Server to be linked with the PED ID.
-id_port <port>	Specifies the PED Server port to be linked with the PED ID.
-logfilename <filename>	Optional. Specifies the log file name to which the logger should log messages.
-loginfo <0 or 1>	Optional. Specifies if the logger should log "info" messages. Set to 0 for no, 1 for yes.
-logwarning <0 or 1>	Optional. Specifies if the logger should log "warning" messages. Set to 0 for no, 1 for yes.
-logerror <0 or 1>	Optional. Specifies if the logger should log "error" messages. Set to 0 for no, 1 for yes.
-logtrace <0 or 1>	Optional. Specifies if the logger should log "trace" messages. Set to 0 for no, 1 for yes.
-maxlogfilesize <size>	Optional. Specifies the maximum log file size in KB.
-locallogger	Optional. Specifies that the Remote PED logger should be used, not the IS logging system.

Example

```
C:\Program Files\Safenet\LunaClient>pedClient –mode setid -id 1234 -id_ip myhostname -id_port 3456
```

pedclient mode show

Queries if PEDclient is currently running and gets details about PEDclient.

Syntax

pedclient mode show [**-admin** <admin port number>] [**-eadmin** <0 or 1>] [**-socketreadtimeout** <int>] [**-socketwritetimeout** <int>] [**-logfilename** <filename>] [**-loginfo** <0 or 1>] [**-logwarning** <0 or 1>] [**-logtrace** <0 or 1>] [**-maxlogfilesize** <size>] [**-locallogger**]

Option	Description
-admin <admin port number>	Optional. Specifies the administration port number to use.
-eadmin <0 or 1>	Optional. Specifies if the administration port is on "localhost" or on the external host name.
-socketreadtimeout <int>	Optional. Specifies the socket read timeout, in seconds.
-socketwritetimeout <int>	Optional. Specifies the socket write timeout, in seconds.
-logfilename <filename>	Optional. Specifies the log file name to which the logger should log messages.
-loginfo <0 or 1>	Optional. Specifies if the logger should log "info" messages. Set to 0 for no, 1 for yes.
-logwarning <0 or 1>	Optional. Specifies if the logger should log "warning" messages. Set to 0 for no, 1 for yes.
-logerror <0 or 1>	Optional. Specifies if the logger should log "error" messages. Set to 0 for no, 1 for yes.
-logtrace <0 or 1>	Optional. Specifies if the logger should log "trace" messages. Set to 0 for no, 1 for yes.
-maxlogfilesize <size>	Optional. Specifies the maximum log file size in KB.
-locallogger	Optional. Specifies that the Remote PED logger should be used, not the IS logging system.

Example

```
C:\Program Files\Safenet\LunaClient>pedClient –mode show
```

pedclient mode start

Starts up the PED Client.

Syntax

pedclient mode start [**-winservice**] [**-eadmin** <0 or 1>] [**-idletimeout** <int>] [**-socketreadtimeout** <int>] [**-socketwritetimeout** <int>] [**-shutdowntimeout** <int>] [**-pstartuptimeout** <int>][**-pshutdowntimeout** <int>] [**-logfilename** <filename>] [**-loginfo** <0 or 1>] [**-logwarning** <0 or 1>] [**-logtrace** <0 or 1>] [**-maxlogfilesize** <size>] [**-locallogger**]

Option	Description
-winservice	Starts PEDclient for Windows service. The standard parameters used for **pedclient mode start** can be used for **pedclient mode start -winservice** as well.
-eadmin <0 or 1>	Optional. Specifies if the administration port is on "localhost" or on the external host name.
-idletimeout <int>	Optional. Specifies the idle connection timeout, in seconds.
-socketreadtimeout <int>	Optional. Specifies the socket read timeout, in seconds.
-socketwritetimeout <int>	Optional. Specifies the socket write timeout, in seconds.
-shutdowntimeout <int>	Optional. Specifies the shutdown timeout for internal services, in seconds.
-pstartuptimeout <int>	Optional. Specifies the startup timeout for the detached process, in seconds.
-pshutdowntimeout <int>	Optional. Specifies the shutdown timeout for the detached process, in seconds.
-logfilename <filename>	Optional. Specifies the log file name to which the logger should log messages.
-loginfo <0 or 1>	Optional. Specifies if the logger should log "info" messages. Set to 0 for no, 1 for yes.
-logwarning <0 or 1>	Optional. Specifies if the logger should log "warning" messages. Set to 0 for no, 1 for yes.
-logerror <0 or 1>	Optional. Specifies if the logger should log "error" messages. Set to 0 for no, 1 for yes.
-logtrace <0 or 1>	Optional. Specifies if the logger should log "trace" messages. Set to 0 for no, 1 for yes.
-maxlogfilesize <size>	Optional. Specifies the maximum log file size in KB.
-locallogger	Optional. Specifies that the Remote PED logger should be used, not the IS logging system.

Example

```
C:\Program Files\Safenet\LunaClient>pedClient –mode start
```

pedclient mode stop

Shuts down PEDclient.

Syntax

pedclient mode stop [**-eadmin** <0 or 1>] [**-socketreadtimeout** <int>] [**-socketwritetimeout** <int>] [**-shutdowntimeout** <int>] [**-pstartuptimeout** <int>][**-pshutdowntimeout** <int>] [**-logfilename** <filename>] [**-loginfo** <0 or 1>] [**-logwarning** <0 or 1>] [**-logtrace** <0 or 1>] [**-maxlogfilesize** <size>] [**-locallogger**]

Option	Description
-eadmin <0 or 1>	Optional. Specifies if the administration port is on "localhost" or on the external host name.
-socketreadtimeout <int>	Optional. Specifies the socket read timeout, in seconds.
-socketwritetimeout <int>	Optional. Specifies the socket write timeout, in seconds.
-shutdowntimeout <int>	Optional. Specifies the shutdown timeout for internal services, in seconds.
-pstartuptimeout <int>	Optional. Specifies the startup timeout for the detached process, in seconds.
-pshutdowntimeout <int>	Optional. Specifies the shutdown timeout for the detached process, in seconds.
-logfilename <filename>	Optional. Specifies the log file name to which the logger should log messages.
-loginfo <0 or 1>	Optional. Specifies if the logger should log "info" messages. Set to 0 for no, 1 for yes.
-logwarning <0 or 1>	Optional. Specifies if the logger should log "warning" messages. Set to 0 for no, 1 for yes.
-logerror <0 or 1>	Optional. Specifies if the logger should log "error" messages. Set to 0 for no, 1 for yes.
-logtrace <0 or 1>	Optional. Specifies if the logger should log "trace" messages. Set to 0 for no, 1 for yes.
-maxlogfilesize <size>	Optional. Specifies the maximum log file size in KB.
-locallogger	Optional. Specifies that the Remote PED logger should be used, not the IS logging system.

Example

```
C:\Program Files\Safenet\LunaClient>pedClient –mode stop
```

pedclient mode testid

Tests a PED ID mapping between a specified PED and PEDserver.

Syntax

pedclient mode testid -id <PED_ID> [**-logfilename** <filename>] [**-loginfo** <0 or 1>] [**-logwarning** <0 or 1>] [**-logerror** <0 or 1>] [**-logtrace** <0 or 1>] [**-maxlogfilesize** <size>] [**-locallogger**]

Option	Description
-id <PED_ID>	Specifies the ID of the PED to be tested.
-logfilename <filename>	Optional. Specifies the log file name to which the logger should log messages.
-loginfo <0 or 1>	Optional. Specifies if the logger should log "info" messages. Set to 0 for no, 1 for yes.
-logwarning <0 or 1>	Optional. Specifies if the logger should log "warning" messages. Set to 0 for no, 1 for yes.
-logerror <0 or 1>	Optional. Specifies if the logger should log "error" messages. Set to 0 for no, 1 for yes.
-logtrace <0 or 1>	Optional. Specifies if the logger should log "trace" messages. Set to 0 for no, 1 for yes.
-maxlogfilesize <size>	Optional. Specifies the maximum log file size in KB.
-locallogger	Optional. Specifies that the Remote PED logger should be used, not the IS logging system.

Example

```
C:\Program Files\Safenet\LunaClient>pedClient –mode testid -id 1234
```

pedserver

Use the **pedserver** commands to manage certificates in PEDserver and the appliance, initiate connections between the PED and HSM, and select the PED for HSM operation.

> **NOTE** The **pedserver** commands are available on Windows only.

To run PEDserver from the command line, you must specify one of the following three options.

Syntax

pedserver

appliance
mode
regen

Option	Description
appliance	Registers or deregisters an appliance, or lists the registered appliances. Applies to server-initiated (peer-to-peer) mode only. See "pedserver appliance" on the next page.
mode	Specifies the mode that the PED Server will be executed in. See "pedserver mode" on page 244.
regen	Regenerates the client certificate. Applies to server-initiated (peer-to-peer) mode only. See "pedserver regen" on page 255.

pedserver appliance

Registers or deregisters an appliance, or lists the registered appliances. These commands apply to PED-initiated mode only.

Syntax

pedserver appliance

delete
list
register

Option	Description
delete	Deregisters an appliance. See "pedserver appliance delete" on the next page.
list	Lists the registered appliances. See "pedserver appliance list" on page 242.
register	Registers an appliance. See "pedserver appliance register" on page 243

pedserver appliance delete

Deregister an appliance certificate from PEDserver.

Syntax

pedserver appliance delete -name <unique name> [**-force**]

Option	Description
-name <unique name>	Specifies the name of the appliance to be deregistered from PEDserver.
-force	Optional parameter. Suppresses any prompts.

Example

```
C:\Program Files\Safenet\LunaClient>pedServer –appliance delete –name hello -force
```

pedserver appliance list

Displays a list of appliances registered with PEDserver.

Syntax

pedserver appliance list

Example

```
C:\Program Files\Safenet\LunaClient>pedServer -appliance list
>
Server Name          IP Address        Port Number     Certificate Common
                                                              Name
_____________________________________________________________________________
abox               192.20.1.23       9697              test2
bbox               192.20.12.34      9696              test1
hello              192.20.1.34       9876              hellocert
```

pedserver appliance register

Register an appliance certificate with PEDserver.

Syntax

pedserver appliance register -name <unique name> **-certificate** <appliance certificate file> **-ip** <appliance server IP address> [**-port** <port number>]

Option	Description
-name <unique name>	Specifies the name of the appliance to be registered to PED Server.
-certificate <appliance certificate file>	Specifies the full path and filename of the certificate that was retrieved from the appliance.
-ip <appliance server IP address>	Specifies the IP address of the appliance server.
-port <port number>	Optional field. Specifies the port number used to connect to the appliance (directly or indirectly according to network configuration). **Range:** 0-65525

Example

```
C:\Program Files\Safenet\LunaClient>pedServer –appliance register –name hello -certificate the-best-appliance.pem -ip 123.321.123.321 –port 9697
```

pedserver mode

Specifies the mode that PEDserver will be executed in.

Syntax

pedserver mode

config
connect
disconnect
show
start
stop

Option	Description
config	Modifies or shows existing configuration file settings. See "pedserver mode config" on the next page.
connect	Connects to the appliance. See "pedserver mode connect" on page 247.
disconnect	Disconnects from the appliance. See "pedserver mode disconnect" on page 248.
show	Queries if PEDserver is currently running, and gets details about PEDserver. See "pedserver mode show" on page 249.
start	Starts PEDserver. See "pedserver mode start" on page 251.
stop	Shuts down PEDserver. See "pedserver mode stop" on page 253

pedserver mode config

Shows and modifies internal PEDserver configuration file settings.

Syntax

pedserver mode config -name <registered appliance name> **-show -set** [**-port** <server port>] [**-set**][**-configfile** <filename>] [**-admin** <admin port number>] [**-eserverport** <0 or 1>] [**-eadmin** <0 or 1>] [**-idletimeout** <int>] [**-socketreadtimeout** <int>] [**-socketwritetimeout** <int>] [**-internalshutdowntimeout** <int>] [**-bgprocessstartuptimeout** <int>] [**-bgprocessshutdowntimeout** <int>] [**-logfilename** <filename>] [**-loginfo** <0 or 1>] [**-logwarning** <0 or 1>] [**-logerror** <0 or 1>] [**-logtrace** <0 or 1>] [**-maxlogfilesize** <size>] [**-pinginterval** <int>] [**-pingtimeout** <int>]

Option	Description
-name <registered appliance name>	Specifies the name of the registered appliance to be configured.
-show	Displays the contents of the PEDserver configuration file.
-set	Updates the PEDserver configuration file to be up to date with other supplied options.
-port <server port>	Optional. Specifies the server port number.
-configfile <filename>	Optional. Specifies which PEDserver configuration file to use.
-admin <admin port number>	Optional. Specifies the administration port number.
-eserverport <0 or 1>	Optional. Specifies if the server port is on "localhost" or listening on the external host name.
-eadmin <0 or 1>	Optional. Specifies if the administration is on "localhost" or listening on the external host name.
-idletimeout <int>	Optional. Specifies the idle connection timeout, in seconds.
-socketreadtimeout <int>	Optional. Specifies the socket read timeout, in seconds.
-socketwritetimeout <int>	Optional. Specifies socket write timeout, in seconds.
-internalshutdowntimeout <int>	Optional. Specifies the shutdown timeout for internal services, in seconds.
-bgprocessstartuptimeout <int>	Optional. Specifies the startup timeout for the detached process, in seconds.

Option	Description
-bgprocessshutdowntimeout <int>	Optional. Specifes the shutdown timeout for the detached process, in seconds.
-logfilename <filename>	Optional. Specifies the log file name to which the logger should log messages.
-loginfo <0 or 1>	Optional. Specifies if the logger should log "info" messages. Set to 0 for no, 1 for yes.
-logwarning <0 or 1>	Optional. Specifies if the logger should log "warning" messages. Set to 0 for no, 1 for yes.
-logerror <0 or 1>	Optional. Specifies if the logger should log "error" messages. Set to 0 for no, 1 for yes.
-logtrace <0 or 1>	Optional. Specifies if the logger should log "trace" messages. Set to 0 for no, 1 for yes.
-maxlogfilesize <size>	Optional. Specifies the maximum log file size in KB.
-pinginterval <int>	Optional. Specifies the time interval between pink commands, in seconds.
-pingtimeout <int>	Optional. Specifies timeout of the ping response, in seconds.

Example

```
C:\Program Files\Safenet\LunaClient>pedServer –mode config -name hellohi -show
```

pedserver mode connect

Connects to the appliance by retrieving information (IP address, port, PEDserver certificate) from the PEDserver configuration file.

If the running mode is legacy, an error is returned. **pedserver mode connect** is not a valid command for legacy connections.

The **connect** command will try connecting to PEDclient 20 times before giving up.

Syntax

pedserver mode connect -name <registered appliance name> [**-configfile** <filename>] [**-logfilename** <filename>] [**-loginfo** <0 or 1>] [**-logwarning** <0 or 1>] [**-logerror** <0 or 1>] [**-logtrace** <0 or 1>] [**-maxlogfilesize** <size>]

Option	Description
-name <registered appliance name>	Specifies the name of the registered appliance to be connected to PEDserver.
-configfile <filename>	Optional. Specifies which PEDserver configuration file to use.
-logfilename <filename>	Optional. Specifies the log file name to which the logger should log messages.
-loginfo <0 or 1>	Optional. Specifies if the logger should log "info" messages. Set to 0 for no, 1 for yes.
-logwarning <0 or 1>	Optional. Specifies if the logger should log "warning" messages. Set to 0 for no, 1 for yes.
-logerror <0 or 1>	Optional. Specifies if the logger should log "error" messages. Set to 0 for no, 1 for yes.
-logtrace <0 or 1>	Optional. Specifies if the logger should log "trace" messages. Set to 0 for no, 1 for yes.
-maxlogfilesize <size>	Optional. Specifies the maximum log file size in KB.

Example

```
C:\Program Files\Safenet\LunaClient>pedServer –mode connect -name hellohi
>Connecting to Luna SA. Please wait....
>Successfully connected to Luna SA.
```

pedserver mode disconnect

Disconnects PEDserver from the appliance.

If the running mode is legacy, an error is returned. **pedserver mode disconnect** is not a valid command for legacy connections.

Termination of the connection may take a few minutes.

Syntax

pedserver mode disconnect -name <registered appliance name> [**-configfile** <filename>] [**-logfilename** <filename>] [**-loginfo** <0 or 1>] [**-logwarning** <0 or 1>] [**-logerror** <0 or 1>] [**-logtrace** <0 or 1>] [**-maxlogfilesize** <size>]

Option	Description
-name <registered appliance name>	Specifies the name of the registered appliance to be disconnected from PEDserver.
-configfile <filename>	Optional. Specifies which PEDserver configuration file to use.
-logfilename <filename>	Optional. Specifies the log file name to which the logger should log messages.
-loginfo <0 or 1>	Optional. Specifies if the logger should log "info" messages. Set to 0 for no, 1 for yes.
-logwarning <0 or 1>	Optional. Specifies if the logger should log "warning" messages. Set to 0 for no, 1 for yes.
-logerror <0 or 1>	Optional. Specifies if the logger should log "error" messages. Set to 0 for no, 1 for yes.
-logtrace <0 or 1>	Optional. Specifies if the logger should log "trace" messages. Set to 0 for no, 1 for yes.
-maxlogfilesize <size>	Optional. Specifies the maximum log file size in KB.

Example

```
C:\Program Files\Safenet\LunaClient>pedServer –mode disconnect –name hellohi
>Connection to Luna SA terminated.
```

pedserver mode show

Queries if PEDserver is currently running, and gets details about PEDserver.

Syntax

pedserver mode show [**-name** <registered appliance name>] [**-configfile** <filename>] [**-logfilename** <filename>] [**-loginfo** <0 or 1>] [**-logwarning** <0 or 1>] [**-logerror** <0 or 1>] [**-logtrace** <0 or 1>] [**-maxlogfilesize** <size>]

Option	Description
-name <registered appliance name>	Specifies the name of the registered appliance to be queried. Applies to server-initiated (peer-to-peer) mode only.
-configfile <filename>	Optional. Specifies which PEDserver configuration file to use.
-logfilename <filename>	Optional. Specifies the log file name to which the logger should log messages.
-loginfo <0 or 1>	Optional. Specifies if the logger should log "info" messages. Set to 0 for no, 1 for yes.
-logwarning <0 or 1>	Optional. Specifies if the logger should log "warning" messages. Set to 0 for no, 1 for yes.
-logerror <0 or 1>	Optional. Specifies if the logger should log "error" messages. Set to 0 for no, 1 for yes.
-logtrace <0 or 1>	Optional. Specifies if the logger should log "trace" messages. Set to 0 for no, 1 for yes.
-maxlogfilesize <size>	Optional. Specifies the maximum log file size in KB.

Example

```
C:\Program Files\Safenet\LunaClient>pedServer -mode show -name hellohi
>Ped Server launched in status mode.
   Server Information:
      Hostname:                          ABC1-123123
      IP:                                192.10.10.123
      Firmware Version:                  2.5.0-1
      PedII Protocol Version:            1.0.1-0
      Software Version:                  1.0.5 (10005)
      Ped2 Connection Status:            Connected
      Ped2 RPK Count                     1
      Ped2 RPK Serial Numbers            (1a123456789a1234)
   Client Information:                   Not Available
   Operating Information:
      Server Port:                       1234
      External Server Interface:         Yes
      Admin Port:                        1235
```

```
      External Admin Interface:           No
      Server Up Time:                     8 (secs)
      Server Idle Time:                   8 (secs) (100%)
      Idle Timeout Value:                 1800 (secs)
      Current Connection Time:            0 (secs)
      Current Connection Idle Time:       0 (secs)
      Current Connection Total Idle Time: 0 (secs) (100%)
      Total Connection Time:              0 (secs)
      Total Connection Idle Time:         0 (secs) (100%)
>Show command passed.
```

pedserver mode start

Starts up PEDserver.

Syntax

pedserver mode start [**-name** <registered appliance name>] [**-ip** <server_IP>] [**-port** <server port>] [**-configfile** <filename>] [**-admin** <admin port number>] [**-eserverport** <0 or 1>] [**-eadmin** <0 or 1>] [**-idletimeout** <int>] [**-socketreadtimeout** <int>] [**-socketwritetimeout** <int>] [**-internalshutdowntimeout** <int>] [**-bgprocessstartuptimeout** <int>] [**-bgprocessshutdowntimeout** <int>] [**-logfilename** <filename>] [**-loginfo** <0 or 1>] [**-logwarning** <0 or 1>] [**-logerror** <0 or 1>] [**-logtrace** <0 or 1>] [**-maxlogfilesize** <size>] [**-pinginterval** <int>] [**-pingtimeout** <int>] [**-force**]

Option	Description
-admin <admin port number>	Optional. Specifies the administration port number.
-bgprocessshutdowntimeout <int>	Optional. Specifes the shutdown timeout for the detached process, in seconds.
-bgprocessstartuptimeout <int>	Optional. Specifies the startup timeout for the detached process, in seconds.
-configfile <filename>	Optional. Specifies which PED Server configuration file to use.
-eadmin <0 or 1>	Optional. Specifies if the administration is on "localhost" or listening on the external host name.
-eserverport <0 or 1>	Optional. Specifies if the server port is on "localhost" or listening on the external host name.
-force	Optional parameter. Suppresses any prompts.
-idletimeout <int>	Optional. Specifies the idle connection timeout, in seconds.
-internalshutdowntimeout <int>	Optional. Specifies the shutdown timeout for internal services, in seconds.
-ip <server_IP>	Optional. Specifies the server listening IP address. When **running pedserver -mode start** on an IPv6 network, you must include this option.
-logerror <0 or 1>	Optional. Specifies if the logger should log "error" messages. Set to 0 for no, 1 for yes.
-logfilename <filename>	Optional. Specifies the log file name to which the logger should log messages.
-loginfo <0 or 1>	Optional. Specifies if the logger should log "info" messages. Set to 0 for no, 1 for yes.

Option	Description
-logtrace <0 or 1>	Optional. Specifies if the logger should log "trace" messages. Set to 0 for no, 1 for yes.
-logwarning <0 or 1>	Optional. Specifies if the logger should log "warning" messages. Set to 0 for no, 1 for yes.
-maxlogfilesize <size>	Optional. Specifies the maximum log file size in KB.
-name <registered appliance name>	
-pinginterval <int>	Optional. Specifies the time interval between pink commands, in seconds.
-pingtimeout <int>	Optional. Specifies timeout of the ping response, in seconds.
-port <server port>	Optional. Specifies the server port number.
-socketreadtimeout <int>	Optional. Specifies the socket read timeout, in seconds.
-socketwritetimeout <int>	Optional. Specifies socket write timeout, in seconds.

Example

```
C:\Program Files\Safenet\LunaClient>pedServer –mode start -name hellohi -force
>Ped Server launched in startup mode.
>Starting background process
>Background process started
>Ped Server Process created, exiting this process.
```

pedserver mode stop

Stops PEDserver.

Syntax

pedserver mode stop [**-name** <registered appliance name>] [**-configfile** <filename>] [**-socketwritetimeout** <int>] [**-internalshutdowntimeout** <int>] [**-bgprocessstartuptimeout** <int>] [**-bgprocessshutdowntimeout** <int>] [**-logfilename** <filename>] [**-loginfo** <0 or 1>] [**-logwarning** <0 or 1>] [**-logerror** <0 or 1>] [**-logtrace** <0 or 1>] [**-maxlogfilesize** <size>]

Option	Description
-name <registered appliance name>	Specifies the name of the registered appliance to be on which PEDserver will be stopped. Applies to server-initiated (peer-to-peer) mode only.
-configfile <filename>	Optional. Specifies which PEDserver configuration file to use.
-socketreadtimeout <int>	Optional. Specifies the socket read timeout, in seconds.
-socketwritetimeout <int>	Optional. Specifies socket write timeout, in seconds.
-internalshutdowntimeout <int>	Optional. Specifies the shutdown timeout for internal services, in seconds.
-bgprocessstartuptimeout <int>	Optional. Specifies the startup timeout for the detached process, in seconds.
-bgprocessshutdowntimeout <int>	Optional. Specifes the shutdown timeout for the detached process, in seconds.
-logfilename <filename>	Optional. Specifies the log file name to which the logger should log messages.
-loginfo <0 or 1>	Optional. Specifies if the logger should log "info" messages. Set to 0 for no, 1 for yes.
-logwarning <0 or 1>	Optional. Specifies if the logger should log "warning" messages. Set to 0 for no, 1 for yes.
-logerror <0 or 1>	Optional. Specifies if the logger should log "error" messages. Set to 0 for no, 1 for yes.
-logtrace <0 or 1>	Optional. Specifies if the logger should log "trace" messages. Set to 0 for no, 1 for yes.
-maxlogfilesize <size>	Optional. Specifies the maximum log file size in KB.

Example

```
C:\Program Files\Safenet\LunaClient>pedServer –mode stop -name hellohi
```

pedserver regen

Regenerates the client certificate. This command is available in server-initiated (peer-to-peer) mode only.

Existing links (PEDserver, NTLS or STC) will not be affected until they are terminated. Afterwards, the user is required to re-register the client certificate to NTLS and PEDserver.

> **NOTE** The **pedserver regen** command should be used only when there is no SafeNet Luna HSM Client installed. When SafeNet Luna HSM Client is installed on the host computer, use the LunaCM command **clientconfig deploy** with the **-regen** option (see "clientconfig deploy" on page 1 in the *LunaCM Command Reference Guide*).

Syntax

pedserver regen -commonname <commonname> [**-force**]

Option	Description
-commonname <commonname>	The client's common name (CN).
-force	Optional parameter. Suppresses any prompts.

Example

```
C:\Program Files\SafeNet\LunaClient>pedServer -regen -commonname win2016_server -force
Ped Server Version 1.0.6 (10006)

Private Key created and written to: C:\Program Files\SafeNet\LunaClient\cert\client\win2016_
serverKey.pem
Certificate created and written to: C:\Program Files\SafeNet\LunaClient\cert\client\win2016_
server.pem

Successfully regenerated the client certificate.
```

CHAPTER 12: Performance Monitoring

An HSM administrator might find it helpful to know how busy the HSM is and at what percentage of its capacity it has been running.

The HSM Information Monitor is a use counter that provides an indication of momentary and cumulative resource usage on the HSM, in the form of a percentage. The HSM firmware tracks the overall time elapsed since the last reset (Up-Time), and the overall time during which the processor was not performing useful work (Idle-Time).

On request, the HSM calculates "Busy-time" over an interval, by subtracting Idle-time for that interval from Up-time for the interval. Then, the load on the processor is calculated as the Busy-time divided by the Up-time, and expressed as a percentage.

You can use the available commands for a single, one-off query, which actually takes an initial reading and then another, five seconds later (the default setting), in order to calculate and show the one-time difference.

You can specify a sampling interval (five seconds is the shortest) and a number of repetitions for an extended view of processor activity/resource usage. The resulting records, showing the time of each measurement, the percentage value at that time, and the difference from the previous measurement, can be output to a file that you import into other tools to analyze and graph the trends.

By watching trends and correlating with what your application is doing, you can:

- Determine the kinds of loads you are placing on the HSM.
- Seek efficiencies in how your applications are coded and configured.
- Plan for expansion or upgrades of your existing HSM infrastructure.
- Plan for upgrades of electrical capacity and HVAC capacity.

Notes about Monitor/Counter Behavior

When performing certain operations the HSM reaches its maximum performance capability before the counter reaches 100%. This occurs because the counter measures the load on the HSM's CPU and the CPU is able to saturate the asymmetric engines and still have capacity to perform other actions.

Also, symmetric cryptographic operations cause the counter to quickly rise to 90% even though there is significant remaining capacity. This behavior occurs because, as the HSM receives more concurrent symmetric commands, its CPU is able to handle them more efficiently (by performing them in bulk) – thus achieving more throughput from the same number of CPU cycles.

See "hsm monitor" on page 1 in the *LunaCM Reference Guide*.

CHAPTER 13: Security in Operation

This section addresses actions and settings with security-related implications.

"Security Effects of Administrative Actions" below

Security Effects of Administrative Actions

Actions that you take, in the course of administering your SafeNet Luna HSM, can have effects, including destruction, on the roles, the spaces, and the contents of your HSM and its application partition(s). It is important to be aware of such consequences before taking action.

Overt Security Actions

Some actions in the administration of the HSM, or of an application partition, are explicitly intended to adjust specific security aspects of the HSM or partition. Examples are:

- Changing a password
- Modifying a policy to make a password or other attribute more stringent than the original setting

Those are discussed in their own sections.

Actions with Security- and Content-Affecting Outcomes

Other administrative events have security repercussions as included effects of the primary action, which could have other intent. Some examples are:

- HSM factory reset
- HSM zeroization
- Change of a destructive policy
- HSM initialization
- HSM firmware rollback
- Application partition initialization

This table lists some major administrative actions that can be performed on the HSM, and compares relevant security-related effects. Use the information in this table to help decide if your contemplated action is appropriate in current circumstances, or if additional preparation (such as backup of partition content, collection of audit data) would be prudent before continuing.

Factory Reset HSM

Domain	Destroyed
HSM SO Role	Destroyed

Partition SO Role	Destroyed
Auditor Role	Destroyed
Partition Roles	Destroyed
HSM or Partition/Contents	HSM/Destroyed
HSM Policies	Reset
RPV	Destroyed
Messaging	You are about to factory reset the HSM. All contents of the HSM will be destroyed. HSM policies will be reset and the remote PED vector will be erased.

Zeroize HSM

Domain	Destroyed
HSM SO Role	Destroyed
Partition SO Role	Destroyed
Auditor Role	Unchanged
Partition Roles	Destroyed
HSM or Partition/Contents	HSM/Destroyed
HSM Policies	Unchanged
RPV	Unchanged
Messaging	You are about to zeroize the HSM. All contents of the HSM will be destroyed. HSM policies, remote PED vector and Auditor left unchanged.

Change Destructive HSM Policy

Domain	Unchanged
HSM SO Role	Unchanged
Partition SO Role	Destroyed

Auditor Role	Unchanged
Partition Roles	Destroyed
HSM or Partition/Contents	HSM/Destroyed
HSM Policies	Unchanged except for new policy
RPV	Unchanged
Messaging	You are about to change a destructive HSM policy. All partitions of the HSM will be destroyed.

HSM Initialize When Zeroized (hard init)

Domain	Destroyed
HSM SO Role	Destroyed
Partition SO Role	Destroyed
Auditor Role	Unchanged
Partition Roles	Destroyed
HSM or Partition/Contents	HSM/Destroyed
HSM Policies	Unchanged
RPV	Unchanged
Messaging	You are about to initialize the HSM. All contents of the HSM will be destroyed.

HSM Initialize From Non-Zeroized State (soft init)

Domain	Unchanged
HSM SO Role	Unchanged
Partition SO Role	Destroyed
Auditor Role	Unchanged
Partition Roles	Destroyed

HSM or Partition/Contents	HSM/Destroyed
HSM Policies	Unchanged
RPV	Unchanged
Messaging	You are about to initialize the HSM that is already initialized. All partitions of the HSM will be destroyed. You are required to provide the current SO password.

HSM Firmware Rollback

Domain	Destroyed
HSM SO Role	Destroyed
Partition SO Role	Destroyed
Auditor Role	Destroyed
Partition Roles	Destroyed
HSM or Partition/Contents	HSM/Destroyed
HSM Policies	Unchanged
RPV	Unchanged
Messaging	WARNING: This operation will rollback your HSM to the previous firmware version !!! (1) This is a destructive operation. (2) You will lose all your partitions. (3) You may lose some capabilities. (4) You must re-initialize the HSM. (5) If the PED use is remote, you must re-connect it.

Partition Initialize When Zeroized (hard init)

Domain	Unchanged
HSM SO Role	Unchanged
Partition SO Role	Destroyed
Auditor Role	Unchanged

Partition Roles	Destroyed
HSM or Partition/Contents	Partition/Destroyed
HSM Policies	Unchanged
RPV	Unchanged
Messaging	You are about to initialize the partition. All contents of the partition will be destroyed.

Partition Initialize From Non-Zeroized State (soft init)

Domain	Unchanged
HSM SO Role	Unchanged
Partition SO Role	Destroyed
Auditor Role	Unchanged
Partition Roles	Destroyed
HSM or Partition/Contents	Partition/Destroyed
HSM Policies	Unchanged
RPV	Unchanged
Messaging	You are about to initialize the partition that is already initialized. All contents of the partition will be destroyed. You are required to provide the current Partition SO password.

Elsewhere

Certain other actions can sometimes cause collateral changes to the HSM, like firmware update. They usually do not affect contents, unless a partition is full and the action changes the size of partitions or changes the amount of space-per-partition that is taken by overhead/infrastructure. These are discussed elsewhere.

CHAPTER 14: Secure Transport Mode

SafeNet HSM 7 units are shipped from the factory in Secure Transport Mode (STM). The purpose of STM is to provide a logical check on the HSM firmware and critical security parameters (such as configuration, keys, policies, roles, etc.) so that the authorized recipient can determine if these have been altered while the HSM was in transit.

Secure Transport Mode overview

The Secure Transport Mode capability provides an additional layer of protection beyond the physical security controls provided by tamper-evident shipping bags.

Gemalto sends customers control validation information in two separate emails prior to shipment:

- **Physical security control validation** - an email containing the serial number of the HSM and the serial number of the associated tamper evident bag that encloses the HSM.
- **Logical control validation** - an email containing the serial number of each HSM in the shipment, along with the STM Random User String and the STM Verification String associated with each HSM.

Customers can use the logical and physical HSM controls to verify that HSMs shipped from the factory have not been modified in transit. The Gemalto shipping procedures are designed to prevent a possible man-in-the-middle attack, as attackers would need unobserved direct access to the HSM while in transit, along with simultaneous possession of both the STM Random User String and the STM Verification String for that HSM.

Gemalto customers can also implement STM when shipping pre-configured HSMs between their office locations or when pre-configured HSMs are to be put into storage. Customers implementing STM have added protection because only the HSM Security Officer can place an initialized HSM into STM, or recover the HSM from STM, further increasing the difficulty of man-in-the-middle attacks.

How does Secure Transport Mode work?

When STM is enabled on the HSM (either at the factory or by customer)

- The HSM generates a random string of 16 characters and presents that as the "Random User String" (suitable for copying and pasting into an e-mail).
- The HSM gathers several sources of internal information reflecting the state of the HSM at that time, including a random nonce value generated for this purpose; the nonce value is not displayed, and never exists outside the HSM.
- The HSM combines these items (the generated Random User String, the HSM state information, and the random nonce value), and produces the "Verification String" (suitable for copying and pasting into an e-mail).
- The HSM then enters Secure Transport Mode, such that only limited operations are allowed until the HSM is brought out of STM.

> The HSM can now be shipped from the factory to customers, or customers can place the HSM into storage or ship securely to another location.
The HSM and the STM strings should not come together until they are in the possession of the intended recipient.

When you recover an HSM from STM:

> The HSM asks for the Random User String (which you received in an e-mail or by other means).

> The HSM gathers the same sources of internal information and combines those with the Random User String that you just provided, and outputs a Verification String.

> **Visually compare** the newly output Verification String with the original Verification String that was sent via e-mail (or other means).

- If the original and the newly generated Verification Strings match, then the HSM has not been used or otherwise altered since STM was enabled.
- If the original and the newly generated Verification Strings fail to match, then there might be a problem with the Random User String - such as an error in the string that was sent, or else an incorrect random user string was entered, or the HSM has been altered somewhere between the original sender and you.

> If the HSM **has not** been altered (original and new Verification Strings match), then you can proceed to recovering the HSM from STM.

> If the HSM might have been altered (original and new Verification Strings are different), then type "quit" at the prompt, and run the **stm recover** command again, to ensure that nothing was incorrectly entered on the first attempt.

> If the Verification strings still do not match:

- type "quit" to leave the HSM in STM, and contact Gemalto Technical Support for further guidance, or
- if you feel that the Verification failure was benign, type "proceed" to release the HSM from Secure Transport Mode, and decide whether
 -- you wish to proceed with using the HSM
or, instead,
 -- you wish to perform factory reset and re-initialize the HSM as a safety precaution before proceeding further.

STM verification email

As part of the delivery process for your new HSM, Thales Client Services will send you an email containing two 16-digit strings: a **Random User String** and a **Verification String**. You require these strings to verify that your HSM has not been altered while in transit.

> **NOTE** If the STM verification process fails due to a lost or incorrect verification string, customers do have the option of proceeding with the recovery of the HSM from STM mode. If the STM verification process fails due to a tamper, customers can also choose to factory-reset the HSM to bring it back to a Factory state, and then re-initialize.

For information about the various tamper events, see "Tamper Events" on page 281.

For command syntax, see "stm" on page 1.

Placing an HSM Into Secure Transport Mode

Only the HSM SO can place an initialized HSM into STM. When the HSM is zeroized, HSM SO log in is not required.

> **CAUTION!** If the HSM contains sensitive key material, ensure that you have a full backup of the HSM contents before proceeding.

To place an HSM into Secure Transport Mode:

1. Log in as the HSM SO.
2. Backup the HSM contents.

   See "Backup and Restore" on page 34 for details.
3. Enter the following command to place the HSM into STM:

   lunacm:>**stm transport**
4. After confirming the action, you are presented with:
   - **Verification String:** <XXXX-XXXX-XXXX-XXXX>
   - **Random User String:** <XXXX-XXXX-XXXX-XXXX>

   Record both strings. They are required to verify that the HSM has not been altered while in STM.

> **CAUTION!** Transmit the verification string and random user string to the receiver of the HSM using a secure method, distinct from the transport of the physical HSM, so that it is not possible for an attacker to have access to both the HSM and the verification codes while the HSM is in STM.

> **CAUTION!** This product uses semiconductors that can be damaged by electro-static discharge (ESD). When handling the device, avoid contact with exposed components, and always use an anti-static wrist strap connected to an earth ground. In rare cases, ESD can trigger a tamper or decommission event on the HSM. If this happens, all existing roles and cryptographic objects are deleted.

Recovering an HSM From Secure Transport Mode

Only the HSM SO can recover an initialized HSM that has been placed into STM. When the HSM is zeroized, HSM SO log in is not required.

New HSMs

New HSMs are shipped from the factory in Secure Transport Mode (STM). You must recover from STM before you can initialize the HSM.

As part of the delivery of your new HSM, you should have received an email from Thales Client Services containing two 16-digit strings:

> Random User String: XXXX-XXXX-XXXX-XXXX

> Verification String: XXXX-XXXX-XXXX-XXXX

To recover an HSM from STM:

1. Ensure that you have the two strings that were presented when the HSM was placed into STM, or that were emailed to you if this is a new HSM.
2. If the HSM is initialized, log in as the HSM SO. If this is a new or zeroized HSM, skip to the next step.
3. Enter the following command to recover from STM, using the random user string that was displayed when the HSM was placed in STM, or that was emailed to you if this is a new HSM.:

   lunacm:> **stm recover -randomuserstring** <XXXX-XXXX-XXXX-XXXX>

   > **NOTE** The random user string is for verification purposes only. If you do not require STM validation, or you wish to bypass the STM validation, you can enter a different string to proceed with the recovery of the HSM from STM mode..

4. You are presented with a verification string:

   If the verification string matches the original verification string, the HSM has not been altered or tampered, and can be safely re-deployed.

   Enter **proceed** to recover from STM.

   If the verification string does not match the original verification string, this might indicate that the HSM has been altered while in transit, or that an incorrect random user string has been entered.

   See "If the verification strings do not match" section below.

If the verification strings do not match:

1. Reconfirm that you have entered the correct random user string for your HSM.
2. If the verification strings still do not match:

   If this is a new HSM, type "quit" to leave the HSM in Secure Transport Mode, and contact Gemalto Technical Support.

   Otherwise,

   - If you feel that the Verification failure was benign, type "proceed" to release the HSM from Secure Transport Mode, and decide to either:
     - proceed with using the HSM
     - perform a factory reset and re-initialize the HSM as a safety precaution before proceeding further.

CHAPTER 15: Slot Numbering and Behavior

Administrative partitions and application partitions are identified as PKCS#11 cryptographic slots in SafeNet utilities, such as LunaCM and multitoken, and for applications that use the SafeNet library.

Order of Occurrence for Different SafeNet Luna HSMs

A host computer with SafeNet Luna HSM Client software and SafeNet libraries installed can have SafeNet Luna HSMs connected in any of three ways:

- PCIe embedded/inserted SafeNet Luna PCIe HSM card (one or multiple HSMs installed - administrative partitions and application partitions are shown separately)
- USB-connected SafeNet Luna USB HSMs (one or multiple - administrative partitions and application partitions are shown separately)
- SafeNet Luna Network HSM application partitions*, registered and connected via NTLS or STC.

Any connected HSM partitions are shown as numbered slots. Slots are numbered from zero or from one, depending on configuration settings (see "Settings Affecting Slot Order" on the next page, below), and on the firmware version of the HSM(s).

* One or multiple application partitions. Administrative partitions on SafeNet Luna Network HSMs are not visible via LunaCM or other client-side tools. Only registered, connected application partitions are visible. The number of visible partitions (up to 100) depends on your model's capabilities. That is, a remote SafeNet Luna Network HSM might support 100 application partitions, but your application and LunaCM will only see partitions that have established certificate-exchange NTLS links with the current Client computer.

In LunaCM, a slot list would normally show:

- SafeNet Luna Network HSM application partitions for which NTLS links are established with the current host, followed by
- SafeNet Luna PCIe HSM cards, followed by
- SafeNet Luna USB HSMs

For SafeNet Luna Network HSM, as seen from a client (via NTLS), only application partitions are visible. The HSM administrative partition of a remote SafeNet Luna Network HSM is never seen by a SafeNet Luna HSM Client. The SafeNet Luna Network HSM slots are listed in the order they are polled, dictated by the entries in the **SafeNet Luna Network HSM** section of the Crystoki.ini / chrystoki.conf file, like this:

```
ServerName00=192.20.17.200
ServerPort00=1792
ServerName01=192.20.17.220
ServerPort01=1793
```

For SafeNet Luna PCIe HSM and SafeNet Luna USB HSM, if you have multiple of either HSM type connected on a single host, then the order in which they appear is the hardware slot number, as discovered by the host computer.

For SafeNet Luna PCIe HSM and SafeNet Luna USB HSM, the HSM administrative slot always appears immediately after the application partition. If no application partition has yet been created, a space is reserved for it, in the slot numbering.

Settings Affecting Slot Order

Settings in the **Presentation** section of the configuration file (Chrystoki.conf for UNIX/Linux, crystoki.ini for Windows) can affect the numbering that the API presents to SafeNet tools (like LunaCM) or to your application.

[Presentation]
ShowUserSlots=<slot>(<serialnumber>)

- Sets starting slot for the identified partition.
- Default, when ShowUserSlots is not specified, is that all available partitions are visible and appear in default order.
- Can be applied, individually, to multiple partitions, by a single entry containing a comma-separated list (with partition serial numbers in brackets):
  ShowUserSlots=1(351970018022), 2(351970018021), 3(351970018020),....
- Affects only PSO partitions (f/w 6.22.0 or newer)
- If multiple partitions on the same HSM are connected to the SafeNet Luna HSM Client host computer, redirecting one of those partitions with ShowUserSlots= causes all the others to disappear from the slot list, unless they are also explicitly re-ordered by the same configuration setting.

ShowAdminTokens=yes

- Default is yes. Admin partitions of local HSMs are visible in a slot listing.
- Remotely connected partitions (SafeNet Luna Network HSM) are not affected by this setting, because NTLS connects only application partitions, not HSM SO (Admin) partitions to clients, so a SafeNet Luna Network HSM SO administrative partition would never be visible in a client-side slot list, regardless.

ShowEmptySlots=1

- Controls how C_GetSlotList - as used by lunacm slot list command, or ckdemo command 14, and by your PKCS#11 application - displays, or does not display unused potential slots, when the number of partitions on an HSM is not at the limit.

OneBaseSlotId=1

- Causes basic slot list to start at slot number 1 (one) instead of default 0 (zero).
  (Any submitted number other than zero is treated as "1". Any letter or other non-numeric character is treated as "0".)

Effects of Settings on Slot List

Say, for example, you have multiple HSMs connected to your host computer (or installed inside), with any combination of firmware 6.22.0 (and newer) or pre-6.22.0 firmware, and no explicit entries exist for slot order in the config file. The defaults prevail and the slot list would start at zero.

If you set OneBaseSlotId=1 in the configuration file, then the slot list starts at "1" instead of at "0". You could set this for personal preference, or according to how your application might expect slot numbering to occur (or if you have existing scripted solutions that depend on slot numbering starting at zero or starting at one). OneBaseSlotId affects the starting number for all slots, regardless of firmware.

If you set ShowUserSlots=20(17923506), then the identified token or HSM or application partition would appear at slot 20, regardless of the locations of other HSMs and partitions.

Effects of New Firmware on Slot Login State

Slots retain login state when current-slot focus changes. You can use the LunaCM command **slot set** to shift focus among slots, and whatever login state existed when you were previously focused on a slot is still in effect when you return to that slot.

CHAPTER 16: SNMP Monitoring

This chapter describes Simple Network Management Protocol (SNMP v3) support for remote monitoring of conditions on a local HSM that might require administrative attention. It contains the following sections:

> "Overview and Installation" below
> "The SafeNet Chrysalis-UTSP MIB" on page 271
> "The SafeNet Luna HSM MIB" on page 272
> "Frequently Asked Questions" on page 280

Overview and Installation

This section provides an overview of the SNMP implementation and describes how to install the SNMP subagent.

MIB

Thales Group provides the following MIBs (management information base) in the SafeNet Luna HSM Client installation package:

MIB Name	Description
CHRYSALIS-UTSP-MIB.txt	Defines SNMP access to information about the SafeNet appliance.
SAFENET-HSM-MIB.txt	Defines SNMP access to information about the SafeNet Luna HSM.
SAFENET-GLOBAL-MIB.txt	Must be found in your system path so that symbols can be resolved.
SAFENET-APPLIANCE-MIB.txt	Reports the software version of SafeNet Luna Network HSM appliance.

Copy all MIBs in **<Luna_HSM_Client_install_dir>/snmp** to the MIB directory on your system. Only the MIBs necessary for SafeNet Luna PCIe HSM and SafeNet Luna USB HSM are included in a client installation.

> **NOTE** Your SNMP application also requires the following standard SNMP MIBs:
> > **SNMPv2-SMI.txt** -- defined in RFC 2578, Section 2
> > **SNMPv2-TC.txt** -- defined in RFC 2579, Section 2

SafeNet SNMP Subagent

We find that most customers choosing to use SNMP already have an SNMP infrastructure in place. Therefore, we provide a subagent that you can install on your managed workstations, and which can point to your agent via the socket created by the agent. This applies to SafeNet Luna USB HSM and SafeNet Luna PCIe HSM - for

SafeNet Luna Network HSM, the subagent is already on the appliance.

The SNMP subagent (luna-snmp) is an AgentX SNMP module that extends an existing SNMP agent with support for SafeNet Luna HSM monitoring. It is an optional component of the SafeNet Luna HSM client installation. The subagent has been tested against net-snmp, but should work with any SNMP agent that supports the AgentX protocol.

To install the SNMP subagent:

After selecting one or more products from the main SafeNet Luna HSM Client installation menu, you are presented with a list of optional components, including the SNMP subagent. It is not selected by default, but can be installed with any product except the SafeNet Luna Network HSM client installed in isolation.

1. In the installation media, go to the appropriate folder for your operating system.
2. Run the installer (install.sh for Linux and UNIX, LunaClient.msi for Windows).
3. Choose the SafeNet products that you wish to install, and include SNMP among your selections. The subagent is installed for any SafeNet product except SafeNet Luna Network HSM in isolation.
4. Proceed to Post-installation configuration.

Post-installation configuration

After the SafeNet Luna HSM client is installed, complete the following steps to configure the SNMP subagent:

1. Copy the SafeNet MIBs from **<install dir>/snmp** to the main SNMP agent's MIB directory. Or copy to another computer (your SNMP computer) if you are not running SNMP from the same computer where SafeNet Luna Client software is installed.
2. If running on Windows, configure the subagent via the file <install dir>/snmp/luna-snmp.conf to point to the AgentX port where the main SNMP agent is listening. The file must then be copied to the same directory as snmpd.conf. (This assumes net-snmp is installed; the setup might differ if you have another agent.)

   If running on a UNIX-based platform, the subagent should work without extra configuration assuming that the primary SNMP agent is listening on the default local socket (/var/agentx/master). You still have the option of editing and using luna-snmp.conf.
3. After configuration is complete, start the agent. Then start the subagent via the service tool applicable to your platform (for example, **service luna-snmp start** on Linux, or start SafeNet SNMP Subagent Service from the services in Windows).

Normally the agent is started first. However, the subagent periodically attempts to connect to the agent until it is successful. The defaults controlling this behavior are listed below. They can be overridden by changing the appropriate entries in **luna-snmp.conf**.

Troubleshooting

If you encounter the following warning:

Warning: Failed to connect to the agentx master agent ([NIL]):

you must enable AgentX support by adding **master agentx** to your SNMPD configuration file. Refer to the man page for **snmpd.conf** for more information.

Configuration Options In the luna-snmp.conf File

Option	Description	Default
agentXSocket [<transport-specifier>:]<transport-address>[,...]	Defines the address to which the subagent should connect. The default on UNIX-based systems is the Unix Domain socket "/var/agentx/master". Another common alternative is tcp:localhost:705. See the section LISTENING ADDRESSES in the snmpd man page for more information about the format of addresses (http://www.net-snmp.org/docs/man/snmpd.html).	The default, for Linux, is "/var/agentx/master". In the file, you can choose to un-comment "tcp:localhost:705" which is most commonly used with Windows.
agentXPingInterval <NUM>	Makes the subagent try to reconnect every <NUM> seconds to the master if it ever becomes (or starts) disconnected.	15
agentXTimeout <NUM>	Defines the timeout period (NUM seconds) for an AgentX request.	1
agentXRetries <NUM>	Defines the number of retries for an AgentX request.	5

The SafeNet Chrysalis-UTSP MIB

> **NOTE** The Chrysalis MIB is the SafeNet MIB for all SafeNet Luna HSM products - the Chrysalis name is retained for historical continuity.

To illustrate accessing data, the command "snmpwalk -v 3 -u admin -l authPriv -a SHA1 -A 12345678 -x AES -X 87654321 myLuna19 private" produced this output:

> CHRYSALIS-UTSP-MIB::hsmOperationRequests.0 = Counter64: 3858380
> CHRYSALIS-UTSP-MIB::hsmOperationErrors.0 = Counter64: 385838
> CHRYSALIS-UTSP-MIB::hsmCriticalEvents.0 = Counter64: 0
> CHRYSALIS-UTSP-MIB::hsmNonCriticalEvents.0 = Counter64: 5
> CHRYSALIS-UTSP-MIB::ntlsOperStatus.0 = INTEGER: up(1)
> CHRYSALIS-UTSP-MIB::ntlsConnectedClients.0 = Gauge32: 0
> CHRYSALIS-UTSP-MIB::ntlsLinks.0 = Gauge32: 0
> CHRYSALIS-UTSP-MIB::ntlsSuccessfulClientConnections.0 = Counter64: 16571615927115620
> CHRYSALIS-UTSP-MIB::ntlsFailedClientConnections.0 = Counter64: 1657161592711562

The various counts are recorded since the last restart.

Item	Description
hsmOperationRequests	The total number of HSM operations that have been requested.
hsmOperationErrors	The total number of HSM operations that have been requested, that have resulted in errors.
hsmCriticalEvents	The total number of critical HSM events that have been detected (Tamper, Decommission, Zeroization, SO creation, or Audit role creation). **NOTE** Not implemented in this release. hsmCriticalEvents always reports 0.
hsmNonCriticalEvents	The total number of NON-critical HSM events that have been detected (any that are not among the critical list, above). **NOTE** Not implemented in this release. hsmNonCriticalEvents always reports 0.
ntlsOperStatus	The current operational status of the NTL service, where the options are: 1 = up, 2 = not running, and 3 = status cannot be determined.
ntlsConnectedClients	The current number of connected clients using NTLS.
ntlsLinks	The current number of links in NTLS - can be multiple per client, depending on processes.
ntlsSuccessfulClientConnections	The total number of successful client connections.
ntlsFailedClientConnections	The total number of UNsuccessful client connections.

The SafeNet Luna HSM MIB

The SAFENET-HSM-MIB defines HSM status information and HSM Partition information that can be viewed via SNMP.

To access tables, use a command like:

```
snmptable  -a SHA  -A snmppass  -u snmpuser -x AES -X snmppass -l authPriv -v 3 192.20.11.59
SAFENET-HSM-MIB::hsmTable
```

The information is defined in tables, as detailed in the following sections.

SNMP Table Updates

The SNMP tables are updated and cached every 60 seconds. Any changes made on the HSM may therefore take up to 60 seconds to be included in the tables. When a query is received to view the tables, the most recent cached version is displayed. If a change you were expecting is not displayed, wait 60 seconds and try again.

NOTE Some values may not get updated automatically, such as the HSM firmware version (hsmFirmwareVersion) following a firmware upgrade. To force an update, restart the SNMP agent.

hsmTable

This table provides a list of all the HSM information on the managed element.

Item	Type	Description	Values
hsmSerialNumber	DisplayString	Serial number of the HSM - used as an index into the tables.	From factory
hsmFirmwareVersion	DisplayString	Version of firmware executing on the HSM.	As found
hsmLabel	DisplayString	Label associated with the HSM.	Provided by SO at init time
hsmModel	DisplayString	Model identifier for the HSM.	From factory
hsmAuthenticationMethod	INTEGER	Authentication mode of the HSM.	unknown(1), -- not known password(2), -- requires passwords pedKeys(3) -- requires PED
hsmRpvInitialized	INTEGER	Remote ped vector initialized flag of the HSM.	notSupported (1), -- rpv not supported uninitialized(2), -- rpv not initialized initialized(3) -- rpv initialized
hsmFipsMode	TruthValue	FIPS 140-2 operation mode enabled flag of the HSM.	Factory set
hsmPerformance	INTEGER	Performance level of the HSM.	
hsmStorageTotalBytes	Unsigned32	Total storage capacity in bytes of the HSM	Factory set
hsmStorageAllocatedBytes	Unsigned32	Number of allocated bytes on the HSM	Calculated
hsmStorageAvailableBytes	Unsigned32	Number of available bytes on the HSM	Calculated

Item	Type	Description	Values
hsmMaximumPartitions	Unsigned32	Maximum number of partitions allowed on the HSM	2, 5, 10, 15, or 20, per license
hsmPartitionsCreated	Unsigned32	Number of partitions created on the HSM	As found
hsmPartitionsFree	Unsigned32	Number of partitions that can still be created on the HSM	Calculated
hsmBackupProtocol	INTEGER	Backup protocol used on the HSM	unknown(1), none(2), cloning(3), keyExport(4)
hsmAdminLoginAttempts	Counter32	Number of failed Administrator login attempts left before HSM zeroized	As found, calculated
hsmAuditRoleInitialized	INTEGER	Audit role is initialized flag	notSupported (0), yes(1), no(2)
hsmManuallyZeroized	TruthValue	Was HSM manually zeroized flag	As found
hsmUpTime	Counter64	Up time in seconds since last HSM reset	Counted
hsmBusyTime	Counter64	Busy time in seconds since the last HSM reset	Calculated
hsmCommandCount	Counter64	HSM commands processed since last HSM reset	Counted

The hsmPartitionTable

This table provides a list of all the partition information on the managed element.

Item	Type	Description	Values
hsmPartitionSerialNumber	DisplayString	Serial number for the partition	Generated
hsmPartitionLabel	DisplayString	Label assigned to the partition	Provided at partition creation
hsmPartitionActivated	TruthValue	Partition activation flag	Set by policy
hsmPartitionStorageTotalBytes	Unsigned32	Total storage capacity in bytes of the partition	Set or calculated at partition creation or re-size

Item	Type	Description	Values
hsmPartitionStorageAllocatedBytes	Unsigned32	Number of allocated (in use) bytes on the partition	Calculated
hsmPartitionStorageAvailableBytes	Unsigned32	Number of avalailable (unused) bytes on the partition	Calculated
hsmPartitionObjectCount	Unsigned32	Number of objects in the partition	Counted

hsmLicenseTable

This table provides a list of all the license information on the managed element. More than one HSM might be connected to a Host, so they are accessed with two indices; the first index identifies the HSM for which the license entry corresponds (hsmSerialNumber), the second is the index for the corresponding license (hsmLicenseID).

Item	Type	Description	Values
hsmLicenseID	DisplayString	License identifier	Set at factory or at capability update
hsmLicenseDescription	DisplayString	License description	Set at factory or at capability update

hsmPolicyTable

This table provides a list of all the HSM policy information on the managed element.

Item	Type	Description	Values
hsmPolicyType	INTEGER	Type of policy	capability(1), policy(2)
hsmPolicyID	Unsigned32	Policy identifier	Numeric value identifies policy and is used as a index into the policy table
hsmPolicyDescription	DisplayString	Description of the policy	Brief text description of what the policy does
hsmPolicyValue	DisplayString	Current value of the policy	Brief text description to show current state/value of policy

hsmPartitionPolicyTable

This table provides a list of all the partition policy information on the managed element.

Item	Type	Description	Values
hsmPartitionPolicyType	INTEGER	Capability or policy	capability(1), policy(2)
hsmPartitionPolicyID	Unsigned32	Policy identifier	Numeric value identifies policy and is used as a index into the policy table
hsmPartitionPolicyDescription	DisplayString	Description of the policy	Brief text description of what the policy does
hsmPartitionPolicyValue	DisplayString	Current value of the policy	Brief text description to show current state/value of policy

hsmClientRegistrationTable

This table provides a list of registered clients.

Item	Type	Description	Values
hsmClientName	DisplayString	Name of the client	Name provided on client cert
hsmClientAddress	DisplayString	Address of the client	IP address of the client
hsmClientRequiresHTL	TruthValue	Flag specifying if HTL required for the client	Flag set at HSM host side to control client access **Note:** HTL is not available in release 7.x. This value will always return **false** for 7.x HSMs.
hsmClientOTTExpiry	INTEGER	OTT expiry time (-1 if not provisioned)	Expiry time, in seconds, for HTL OneTimeToken (range is 0-3600); -1 indicates not provisioned, 0 means never expires **Note:** HTL is not available in release 7.x. This value will always return **-1** for 7.x HSMs.

hsmClientPartitionAssignmentTable

This table provides a list of assigned partitions for a given client.

Item	Type	Description	Values
hsmClientHsmSerialNumber	DisplayString	Index into the HSM table	--
hsmClientPartitionSerialNumber DisplayString	DisplayString	Index into the Partition Table	--

SNMP output compared to SafeNet tools output

For comparison, the following shows LunaCM or LunaSH command outputs that provide HSM information equivalent to the SNMP information depicted in the tables above (from the HSM MIB).

HSM Information

At the HSM level the information in the outputs of **hsm show** and **hsm showpolicies** and **hsm displaylicenses** includes the following:

- SW Version
- FW Version
- HSM label
- Serial #
- HW Model
- Authentication Method
- RPV state
- FIPS mode
- HSM total storage space (bytes)
- HSM used storage space (bytes)
- HSM free storage space (bytes)
- Performance level
- Max # of partitions
- # of partitions created
- # of free partitions
- Policies as shown below:

```
lunash:>hsm showpolicies


   HSM Label:   myLunaHSM
   Serial #:    66331
   Firmware:    7.3.0

   The following capabilities describe this HSM, and cannot be altered
   except via firmware or capability updates.

   Description                           Value
   ===========                           =====
   Enable PIN-based authentication       Allowed
   Enable PED-based authentication       Disallowed
   Performance level                     15
   Enable domestic mechanisms & key sizes   Allowed
   Enable masking                        Disallowed
   Enable cloning                        Allowed
   Enable full (non-backup) functionality   Allowed
   Enable non-FIPS algorithms            Allowed
   Enable SO reset of partition PIN      Allowed
   Enable network replication            Allowed
   Enable Korean Algorithms              Disallowed
```

```
   FIPS evaluated                          Disallowed
   Manufacturing Token                     Disallowed
   Enable forcing user PIN change          Allowed
   Enable portable masking key             Allowed
   Enable partition groups                 Disallowed
   Enable remote PED usage                 Disallowed
   HSM non-volatile storage space          33554432
   Enable unmasking                        Allowed
   Maximum number of partitions            100
   Enable Single Domain                    Disallowed
   Enable Unified PED Key                  Disallowed
   Enable MofN                             Disallowed
   Enable small form factor backup/restore Disallowed
   Enable Secure Trusted Channel           Allowed
   Enable decommission on tamper           Allowed
   Enable partition re-initialize          Disallowed
   Enable low level math acceleration      Allowed
   Enable Fast-Path                        Disallowed
   Allow Disabling Decommission            Allowed
   Enable Tunnel Slot                      Disallowed
   Enable Controlled Tamper Recovery       Allowed
   Enable Partition Utilization Metrics    Allowed


   The following policies are set due to current configuration of
   this HSM and cannot be altered directly by the user.

   Description                             Value
   ===========                             =====
   PIN-based authentication                True


   The following policies describe the current configuration of
   this HSM and may be changed by the HSM Administrator.
   Changing policies marked "destructive" will erase all HSM partitions
   on the HSM.

   IMPORTANT NOTE: Changing policy 46 (Disable Decommission) will erase
   all partitions AND zeroize your HSM.

   Description                             Value       Code     Destructive
   ===========                             =====       ====     ===========
   Allow cloning                           On          7        Yes
   Allow non-FIPS algorithms               On          12       Yes
   SO can reset partition PIN              Off         15       Yes
   Allow network replication               On          16       No
   Force user PIN change after set/reset   On          21       No
   Allow offboard storage                  On          22       Yes
   Allow unmasking                         On          30       No
   Current maximum number of partitions    100         33       No
   Allow Secure Trusted Channel            Off         39       No
   Decommission on tamper                  Off         40       Yes
   Allow low level math acceleration       On          43       No
   Disable Decommission                    Off         46       Yes
   Do Controlled Tamper Recovery           On          48       No
   Allow Partition Utilization Metrics     Off         49       No


Command Result : 0 (Success)
```

Partition Information

At the application partition level, the information in the outputs of **partition show** and **partition showpolicies** includes the following:

- Partition Name
- Partition Serial #
- Activation State
- AutoActivation State
- Partition total storage space (bytes)
- Partition used storage space (bytes)
- Partition free storage space (bytes)
- Partition Object Count
- Partition policies from the **partition showpolicies** command:

```
lunacm:> partition showpolicies
        Partition Capabilities
                 0: Enable private key cloning : 1
                 1: Enable private key wrapping : 1
                 2: Enable private key unwrapping : 1
                 3: Enable private key masking : 0
                 4: Enable secret key cloning : 1
                 5: Enable secret key wrapping : 1
                 6: Enable secret key unwrapping : 1
                 7: Enable secret key masking : 0
                10: Enable multipurpose keys : 1
                11: Enable changing key attributes : 1
                15: Allow failed challenge responses : 1
                16: Enable operation without RSA blinding : 1
                17: Enable signing with non-local keys : 1
                18: Enable raw RSA operations : 1
                20: Max failed user logins allowed : 10
                21: Enable high availability recovery : 1
                22: Enable activation : 0
                23: Enable auto-activation : 0
                25: Minimum pin length (inverted: 255 - min) : 248
                26: Maximum pin length : 255
                28: Enable Key Management Functions : 1
                29: Enable RSA signing without confirmation : 1
                31: Enable private key unmasking : 1
                32: Enable secret key unmasking : 1
                33: Enable RSA PKCS mechanism : 1
                34: Enable CBC-PAD (un)wrap keys of any size : 1
                37: Enable Secure Trusted Channel : 1
                39: Enable  Start/End Date Attributes : 1

        Partition Policies
                 0: Allow private key cloning : 1
                 1: Allow private key wrapping : 0
                 2: Allow private key unwrapping : 1
                 3: Allow private key masking : 0
                 4: Allow secret key cloning : 1
                 5: Allow secret key wrapping : 1
                 6: Allow secret key unwrapping : 1
                10: Allow multipurpose keys : 1
                11: Allow changing key attributes : 1
                15: Ignore failed challenge responses : 1
```

```
                16: Operate without RSA blinding : 1
                17: Allow signing with non-local keys : 1
                18: Allow raw RSA operations : 1
                20: Max failed user logins allowed : 10
                21: Allow high availability recovery : 1
                22: Allow activation : 0
                23: Allow auto-activation : 0
                25: Minimum pin length (inverted: 255 - min) : 248
                26: Maximum pin length : 255
                28: Allow Key Management Functions : 1
                29: Perform RSA signing without confirmation : 1
                31: Allow private key unmasking : 1
                32: Allow secret key unmasking : 1
                33: Allow RSA PKCS mechanism : 1
                34: Allow CBC-PAD (un)wrap keys of any size : 1
                37: Force Secure Trusted Channel : 0
                39: Allow Start/End Date Attributes : 0

Command Result : No Error
```

Frequently Asked Questions

This section provides additional information by answering questions that are frequently asked by our customers.

We want to use SNMP to remotely monitor and manage our installation – why do you not support such standard SNMP traps as CPU and Memory exhaustion?

Those sorts of traps were specifically excluded because they can be used to establish a covert channel (an illicit signaling channel that can be used to communicate from a high assurance “area” to a lower assurance one in an effort to circumvent the security policy). Resource exhaustion events/alerts are the oldest known form of covert channel signaling. Exercise care with any HSM product that does allow such traps - what other basic security holes might be present?

CHAPTER 17: Tamper Events

SafeNet Luna PCIe HSMs detect hardware anomalies (such as card over-temperature) and physical events (such as card removal or chassis intrusion), and register them as tamper events. A tamper event is considered a security breach, and effectively locks the HSM.

If **Policy 48: Do Controlled Tamper Recovery** is enabled (the default), the HSM SO must clear the tamper condition before the HSM is reset, to return the HSM to normal operation (see "HSM Capabilities and Policies" on page 82). While the HSM is in the tamper condition, only the subset of LunaCM commands required to view the HSM status or clear the tamper condition are available. For PED-authenticated HSMs, the cached PED key data that allows activation is zeroized, and activation is disabled. When an HSM is in the tamper state, only the HSM SO is able to log in to the HSM.

You can enable **Policy 40: Decommission on Tamper** to decommission the HSM when a tamper event occurs, so that partitions and roles are deleted from the HSM. By default, **Policy 40: Decommission on Tamper** is disabled, and the contents of the HSM are not affected by the tamper event.

If both policies are disabled, the HSM sends a warning when a tamper event occurs but does not make partition data inaccessible. We do not recommend disabling both policies.

If both policies are enabled, the HSM SO role is deleted when a tamper event occurs, so you do not need to log in this role to clear the tamper condition.

There are several conditions that can result in a tamper event. The type of tamper event is indicated by the **HSM Status** field in the output of the LunaCM **slot list** command. The status also indicates whether the tamper event requires an HSM reset in addition to a tamper clear.

> **NOTE** A tamper event resets the HSM hardware, including the PCIe logic. This prevents the HSM from reporting any statuses, including the cause of the tamper condition. The only thing which is detected in this case is k7pf0: ALM0015: PCIe Link Failure. The HSM must be rebooted before the cause of the tamper event can be reported.

Tamper event	Response
Chassis intrusion (requires chassis connector to card tamper header)	Halt the HSM. Deactivate activated partitions. Decommission the HSM if **policy 40: Decommission on Tamper** is enabled.
Card removal	Halt the HSM. Deactivate activated partitions. Decommission the HSM if **policy 40: Decommission on Tamper** is enabled.

Tamper event	Response
Over/under temperature	Halt the HSM. Deactivate activated partitions. Decommission the HSM if **policy 40: Decommission on Tamper** is enabled. Warnings are logged for mild over/under temperature events. Warnings are self-clearing if the condition is resolved.
Over/under voltage	Halt the HSM. Deactivate activated partitions. Decommission the HSM if **policy 40: Decommission on Tamper** is enabled. Warnings are logged for mild over/under voltage events. Warnings are self-clearing if the condition is resolved.
Battery removal/depletion	Halt the HSM. Deactivate activated partitions. Decommission the HSM. Warnings are logged for low battery conditions.

Recovering from a Tamper Event

How you recover from a tamper event depends on how the following HSM policies are set. See "HSM Capabilities and Policies" on page 82 for more information:

Policy 40: Decommission on tamper	If enabled, the HSM is decommissioned when a tamper event occurs. You must clear the tamper condition before you can re-initialize the HSM SO, re-create your partitions, restore the partition contents from backup, and re-initialize the partition roles (Partition SO, Crypto Officer, and Crypto User, and Audit, as relevant).
Policy 48: Do Controlled Tamper Recovery	If enabled, the tamper condition that halted the HSM must be cleared by the HSM SO (by issuing the **tamper clear** command), before the HSM can be reset to resume normal operations.

Activation and auto-activation is disabled on tamper

If you are using activation or auto-activation on your PED-authenticated partitions, it is disabled when a tamper is detected, or if any uncleared tamper conditions are detected on reboot. See "Activation and Auto-Activation on PED-Authenticated Partitions" on page 178 and "Partition Capabilities and Policies" on page 87 for more information.

To recover from a tamper

1. View the output of the **slot list** command (displayed by default on login). The reason for the tamper is indicated by the **HSM Status** field. You can also use the **hsm tampershow** command to display the last tamper event.

> **NOTE** The **slot list** and **hsm tampershow** commands only show the last tamper event, even if several tampers have occurred. To view a complete list of the tamper events that have occurred on the HSM, use the **lunadiag** utility. See "lunadiag" on page 1 in the *Utilities Guide* for more information.

2. Resolve the issue(s) that caused the tamper event.
3. If **Policy 48: Do Controlled Tamper Recovery** is enabled, clear the tamper condition. Otherwise, go to the next step:

   lunacm:> **hsm tamperclear**
4. If the tamper message indicates that a reset is required, exit LunaCM and use the **lunareset** command to reset the HSM. See "Lunareset" on page 1 in the *Utilities Reference Guide* for more information:

   lunareset <device>
5. Verify that all tampers have been cleared:

   lunacm:> **hsm tampershow**
6. If the HSM was decommissioned as a result of the tamper, you must re-create your partitions, re-initialize the partition roles (Partition SO, Crypto Officer, and Crypto User, and Audit as relevant), and restore the partition contents from backup. See the following sections in the Configuration Guide .
   a. To re-create your partitions, see "Creating an Application Partition on the HSM" on page 1.
   b. Re-initialize the partition roles. See "Creating an Application Partition on the HSM" on page 1.
   c. To restore the partition contents from backup, see "Backup and Restore" on page 34.
7. If the **Policy 22: Allow Activation** and/or **Policy 23: Allow AutoActivation** are enabled on your PED-authenticated partitions, the CO and CU (if enabled) must log in to reactivate those roles:

   lunacm:> **role login -name** <role>

CHAPTER 18: Troubleshooting

This chapter lists the HSM error codes and offers troubleshooting tips for some common issues. It contains the following sections:

> "General Troubleshooting Tips" below
> "System Operational and Error Messages" below
> "Keycard and Token Return Codes" on page 286
> "Library Codes" on page 304
> "Vendor-Defined Return Codes" on page 308

General Troubleshooting Tips

Here are just a few quick things to check if you are experiencing problems:

> If you see an apparent 'hang' condition, connect and check the PED - it may be waiting for a PED action.
> Check if you allowed the PED to time out, or if you started a command that needed PED action while the PED was not connected. You will need to re-issue the failed command after re-inserting the token, and pay attention to the PED.
> If RSA signing seems slow, check the Capabilities and Policies to ensure that Confirmation (policy #29) is switched off - if your security policy demands that signing operations must be verified on the HSM, then expect almost a 50% performance reduction.
> If you perform a Restore from Backup operation and some or all of the objects are shown with an error message like "LUNA_RET_SM_ACCESS_DOES_NOT_VALIDATE", you might have interrupted the restore operation (even a **partition contents** command could have this effect). Re-issue the Restore command, ensuring that no other commands are run against the partition while the operation is in progress - if other persons might be using their own SSH sessions to access the appliance, it might be best to disconnect the network cable and perform your restore operation from the local (serial) console.

System Operational and Error Messages

Extra slots that say "token not present"?

This happens for two reasons:

> PKCS#11 originated in a world of software cryptography, which only later acknowledged the existence of Hardware Security Modules, so initially it did not have the concept of physically removable crypto slots. PKCS#11 requires a static list of slots when an application starts. The cryptographic "token" can be inserted into, or removed from a slot dynamically (by a user), for the duration of the application.

> When the token is inserted, the running application must be able to detect that token. When the token is removed, the running application gets "token not present". Because we allow for the possibility of backup, we routinely declare 'place-holder' slots that might later be filled by a physical SafeNet Luna USB HSM or a SafeNet Luna Backup HSM.

In the Chrystoki.conf file (or the Windows crystoki.ini file), for SafeNet Luna USB HSM, you can remove the empty slots by modifying the CardReader entry, like this:

```
CardReader = {
 LunaG5Slots=0;
}
```

For SafeNet Luna Network HSM, which has its configuration file internal to the appliance, and not directly accessible for modification, you cannot change the default cryptographic slot allotments.

Error: 'hsm update firmware' failed. (10A0B : LUNA_RET_OPERATION_RESTRICTED) when attempting to perform hsm update firmware

You must ensure that STM is disabled before you run the firmware update.

Also, as with any update, you should backup any important HSM contents before proceeding.

KR_ECC_POINT_INVALID Error when decrypting a file encrypted from BSAFE through ECIES using ECC key with any of the curves from the x9_t2 section

As indicated on the BSAFE web site, they support only the NIST-approved curves (prime, Binary, and Koblitz). That includes most/all the curves from test items 0 through 37 in CK Demo: the "secp", "X9_62_prime", and "sect" curves.

The X9.62 curves that are failing in this task are X9.62 binary/char2 curves which do not appear to be supported by BSAFE. So, you appear to be encountering a BSAFE limitation and not a SafeNet Luna HSM problem.

Slow/interrupted response from the HSM, and the "hsm show" command shows LUNA_RET_SM_SESSION_REALLOC_ERROR

```
   Appliance Details:
   ==================
   Software Version:                7.0.0
Error:  'hsm show' failed. (310102 : LUNA_RET_SM_SESSION_REALLOC_ERROR)

Command Result : 65535 (Luna Shell execution)
```

The error LUNA_RET_SM_SESSION_REALLOC_ERROR means the HSM cannot expand the session table.

The HSM maintains a table for all of the open sessions. For performance reasons, the table is quite small initially. As sessions are opened (and not closed) the table fills up. When the table gets full, the HSM tries to expand the table. If there is not enough available RAM to grow the table, this error is returned.

RAM can be used up by an application that creates and does not delete a large number of session objects, as well as by an application that opens and fails to close a large number of sessions.

The obvious solution is proper housekeeping. Your applications must clean up after themselves, by closing sessions that are no longer in use - this deletes session objects associated with those sessions. If your application practice is to have long-lived sessions, and to open many objects in a given session, then your application should explicitly delete those session objects as soon as each one is no longer necessary.

By far, we see more of the former problem - abandoned sessions - and very often in conjunction with Java-based applications. Proper garbage collection includes deleting session objects when they are no longer useful, or simply closing sessions as soon as they are not required. Formally closing a session (or stopping/restarting the HSM) deletes all session objects within each affected session. These actions keep the session table small, so it uses the least possible HSM volatile memory.

Low Battery Message

The K7 HSM card, used in the SafeNet Luna Network HSM and SafeNet Luna PCIe HSM products, is equipped with a non-replaceable battery that is expected to last the life of the product. If you notice a log message or other warning about 'battery low', or similar, contact SafeNet Technical Support.

Keycard and Token Return Codes

The following table summarizes HSM error codes (last updated for firmware 7.0.1):

HSM Error	Hex Code	PKCS#11 or SFNT Defined CKR Error
LUNA_RET_OK	0x00000000	CKR_OK
LUNA_RET_CANCEL	0x00010000	CKR_CANCEL
LUNA_RET_FLAGS_INVALID	0x00040000	CKR_FLAGS_INVALID, removed from v2.0
LUNA_RET_TOKEN_NOT_PRESENT	0x00E00000	CKR_TOKEN_NOT_PRESENT
LUNA_RET_FORMER_INVALID_ENTRY_TYPE	0x00300130	CKR_DEVICE_ERROR
LUNA_RET_SP_TX_ERROR	0x00300131	CKR_DEVICE_ERROR
LUNA_RET_SP_RX_ERROR	0x00300132	CKR_DEVICE_ERROR
LUNA_RET_PED_ID_INVALID	0x00300140	CKR_DEVICE_ERROR
LUNA_RET_PED_UNSUPPORTED_PROTOCOL	0x00300141	CKR_DEVICE_ERROR
LUNA_RET_PED_UNPLUGGED	0x00300142	CKR_PED_UNPLUGGED
LUNA_RET_PED_ERROR	0x00300144	CKR_DEVICE_ERROR

HSM Error	Hex Code	PKCS#11 or SFNT Defined CKR Error
LUNA_RET_PED_UNSUPPORTED_CRYPTO_PROTOCOL	0x00300145	CKR_DEVICE_ERROR
LUNA_RET_PED_DEK_INVALID	0x00300146	CKR_DEVICE_ERROR
LUNA_RET_PED_CLIENT_NOT_RUNNING	0x00300147	CKR_PED_CLIENT_NOT_RUNNING
LUNA_RET_CL_ALIGNMENT_ERROR	0x00300200	CKR_DEVICE_ERROR
LUNA_RET_CL_QUEUE_LOCATION_ERROR	0x00300201	CKR_DEVICE_ERROR
LUNA_RET_CL_QUEUE_OVERLAP_ERROR	0x00300202	CKR_DEVICE_ERROR
LUNA_RET_CL_TRANSMISSION_ERROR	0x00300203	CKR_DEVICE_ERROR
LUNA_RET_CL_NO_TRANSMISSION	0x00300204	CKR_DEVICE_ERROR
LUNA_RET_CL_COMMAND_MALFORMED	0x00300205	CKR_DEVICE_ERROR
LUNA_RET_CL_MAILBOXES_NOT_AVAILABLE	0x00300206	CKR_DEVICE_ERROR
LUNA_RET_MM_NOT_ENOUGH_MEMORY	0x00310000	CKR_DEVICE_ERROR
LUNA_RET_MM_INVALID_HANDLE	0x00310001	CKR_DEVICE_ERROR
LUNA_RET_MM_USAGE_ALREADY_SET	0x00310002	CKR_DEVICE_ERROR
LUNA_RET_MM_ACCESS_OUTSIDE_ALLOCATION_RANGE	0x00310003	CKR_DEVICE_ERROR
LUNA_RET_MM_INVALID_USAGE	0x00310004	CKR_DEVICE_ERROR
LUNA_RET_MM_ITERATOR_PAST_END	0x00310005	CKR_DEVICE_ERROR
LUNA_RET_MM_FATAL_ERROR	0x00310006	CKR_DEVICE_ERROR
LUNA_RET_TEMPLATE_INCOMPLETE	0x00D00000	CKR_TEMPLATE_INCOMPLETE
LUNA_RET_TEMPLATE_INCONSISTENT	0x00D10000	CKR_TEMPLATE_INCONSISTENT*
LUNA_RET_ATTRIBUTE_TYPE_INVALID	0x00120000	CKR_ATTRIBUTE_TYPE_INVALID
LUNA_RET_ATTRIBUTE_VALUE_INVALID	0x00130000	CKR_ATTRIBUTE_VALUE_INVALID

HSM Error	Hex Code	PKCS#11 or SFNT Defined CKR Error
LUNA_RET_ATTRIBUTE_READ_ONLY	0x00100000	CKR_ATTRIBUTE_READ_ONLY
LUNA_RET_ATTRIBUTE_SENSITIVE	0x00110000	CKR_ATTRIBUTE_SENSITIVE
LUNA_RET_OBJECT_HANDLE_INVALID	0x00820000	CKR_OBJECT_HANDLE_INVALID
LUNA_RET_MAX_OBJECT_COUNT	0x00820001	CKR_MAX_OBJECT_COUNT_ EXCEEDED
LUNA_RET_ATTRIBUTE_NOT_FOUND	0x00120010	CKR_ATTRIBUTE_TYPE_INVALID
LUNA_RET_CAN_NOT_CREATE_SECRET_KEY	0x00D10011	CKR_TEMPLATE_INCONSISTENT
LUNA_RET_CAN_NOT_CREATE_PRIVATE_KEY	0x00D10012	CKR_TEMPLATE_INCONSISTENT
LUNA_RET_SECRET_KEY_MUST_BE_SENSITIVE	0x00130013	CKR_ATTRIBUTE_VALUE_INVALID
LUNA_RET_SECRET_KEY_MUST_HAVE_ SENSITIVE_ATTRIBUTE	0x00D00014	CKR_TEMPLATE_INCOMPLETE
LUNA_RET_PRIVATE_KEY_MUST_BE_SENSITIVE	0x00130015	CKR_ATTRIBUTE_VALUE_INVALID
LUNA_RET_PRIVATE_KEY_MUST_HAVE_ SENSITIVE_ATTRIBUTE	0x00D00016	CKR_TEMPLATE_INCOMPLETE
LUNA_RET_SIGNING_KEY_MUST_BE_LOCAL	0x00680001	CKR_KEY_FUNCTION_NOT_ PERMITTED
LUNA_RET_MULTI_FUNCTION_KEYS_NOT_ ALLOWED	0x00D10018	CKR_TEMPLATE_INCONSISTENT
LUNA_RET_CAN_NOT_CHANGE_KEY_FUNCTION	0x00100019	CKR_ATTRIBUTE_READ_ONLY
LUNA_RET_KEY_SIZE_RANGE	0x00620000	CKR_KEY_SIZE_RANGE
LUNA_RET_KEY_TYPE_INCONSISTENT	0x00630000	CKR_KEY_TYPE_INCONSISTENT
LUNA_RET_KEY_INVALID_FOR_OPERATION	0x00630001	CKR_KEY_TYPE_INCONSISTENT
LUNA_RET_KEY_PARITY	0x00630002	CKR_KEY_TYPE_INCONSISTENT
LUNA_RET_KEY_UNEXTRACTABLE	0x006a0000	CKR_KEY_UNEXTRACTABLE
LUNA_RET_KEY_EXTRACTABLE	0x006a0001	KR_KEY_UNEXTRACTABLE

HSM Error	Hex Code	PKCS#11 or SFNT Defined CKR Error
LUNA_RET_KEY_INDIGESTIBLE	0x00670000	CKR_KEY_INDIGESTIBLE
LUNA_RET_KEY_NOT_WRAPPABLE	0x00690000	CKR_KEY_NOT_WRAPPABLE
LUNA_RET_KEY_NOT_UNWRAPPABLE	0x00690001	CKR_KEY_NOT_WRAPPABLE
LUNA_RET_ARGUMENTS_BAD	0x00070000	CKR_ARGUMENTS_BAD
LUNA_RET_INVALID_ENTRY_TYPE	0x00070001	CKR_INVALID_ENTRY_TYPE
LUNA_RET_DATA_INVALID	0x00200000	CKR_DATA_INVALID
LUNA_RET_SM_DATA_INVALID	0x00200002	CKR_DATA_INVALID
LUNA_RET_NO_RNG_SEED	0x00200015	CKR_DATA_INVALID
LUNA_RET_FUNCTION_NOT_SUPPORTED	0x00540000	CKR_FUNCTION_NOT_ SUPPORTED
LUNA_RET_NO_OFFBOARD_STORAGE	0x00540001	CKR_FUNCTION_NOT_ SUPPORTED
LUNA_RET_CL_COMMAND_NON_BACKUP	0x00540002	CKR_FUNCTION_NOT_ SUPPORTED
LUNA_RET_BUFFER_TOO_SMALL	0x01500000	CKR_BUFFER_TOO_SMALL
LUNA_RET_DATA_LEN_RANGE	0x00210000	CKR_DATA_LEN_RANGE
LUNA_RET_GENERAL_ERROR	0x00050000	CKR_GENERAL_ERROR
LUNA_RET_DEVICE_ERROR	0x00300000	CKR_DEVICE_ERROR
LUNA_RET_UNKNOWN_COMMAND	0x00300001	CKR_FUNCTION_NOT_ SUPPORTED
LUNA_RET_TOKEN_LOCKED_OUT	0x00300002	CKR_PIN_LOCKED
LUNA_RET_RNG_ERROR	0x00300003	CKR_DEVICE_ERROR
LUNA_RET_DES_SELF_TEST_FAILURE	0x00300004	CKR_DEVICE_ERROR
LUNA_RET_CAST_SELF_TEST_FAILURE	0x00300005	CKR_DEVICE_ERROR

HSM Error	Hex Code	PKCS#11 or SFNT Defined CKR Error
LUNA_RET_CAST3_SELF_TEST_FAILURE	0x00300006	CKR_DEVICE_ERROR
LUNA_RET_CAST5_SELF_TEST_FAILURE	0x00300007	CKR_DEVICE_ERROR
LUNA_RET_MD2_SELF_TEST_FAILURE	0x00300008	CKR_DEVICE_ERROR
LUNA_RET_MD5_SELF_TEST_FAILURE	0x00300009	CKR_DEVICE_ERROR
LUNA_RET_SHA_SELF_TEST_FAILURE	0x0030000a	CKR_DEVICE_ERROR
LUNA_RET_RSA_SELF_TEST_FAILURE	0x0030000b	CKR_DEVICE_ERROR
LUNA_RET_RC2_SELF_TEST_FAILURE	0x0030000c	CKR_DEVICE_ERROR
LUNA_RET_RC4_SELF_TEST_FAILURE	0x0030000d	CKR_DEVICE_ERROR
LUNA_RET_RC5_SELF_TEST_FAILURE	0x0030000e	CKR_DEVICE_ERROR
LUNA_RET_SO_LOGIN_FAILURE_THRESHOLD	0x0030000f	CKR_SO_LOGIN_FAILURE_ THRESHOLD
LUNA_RET_RNG_SELF_TEST_FAILURE	0x00300010	CKR_DEVICE_ERROR
LUNA_RET_SM_UNKNOWN_COMMAND	0x00300011	CKR_DEVICE_ERROR
LUNA_RET_UM_TSN_MISSING	0x00300012	CKR_DEVICE_ERROR
LUNA_RET_SM_TSV_MISSING	0x00300013	CKR_DEVICE_ERROR
LUNA_RET_SM_UNKNOWN_TOSM_STATE	0x00300014	CKR_DEVICE_ERROR
LUNA_RET_DSA_PARAM_GEN_FAILURE	0x00300015	CKR_DEVICE_ERROR
LUNA_RET_DSA_SELF_TEST_FAILURE	0x00300016	CKR_DEVICE_ERROR
LUNA_RET_SEED_SELF_TEST_FAILURE	0x00300017	CKR_DEVICE_ERROR
LUNA_RET_AES_SELF_TEST_FAILURE	0x00300018	CKR_DEVICE_ERROR
LUNA_RET_FUNCTION_NOT_SUPPORTED_BY_ HARDWARE	0x00300019	CKR_DEVICE_ERROR
LUNA_RET_HAS160_SELF_TEST_FAILURE	0x0030001a	CKR_DEVICE_ERROR

HSM Error	Hex Code	PKCS#11 or SFNT Defined CKR Error
LUNA_RET_KCDSA_PARAM_GEN_FAILURE	0x0030001b	CKR_DEVICE_ERROR
LUNA_RET_KCDSA_SELF_TEST_FAILURE	0x0030001c	CKR_DEVICE_ERROR
LUNA_RET_HSM_INTERNAL_BUFFER_TOO_ SMALL	0x0030001d	CKR_DEVICE_ERROR
LUNA_RET_COUNTER_WRAPAROUND	0x0030001e	CKR_DEVICE_ERROR
LUNA_RET_TIMEOUT	0x0030001f	CKR_TIMEOUT
LUNA_RET_NOT_READY	0x00300020	CKR_DEVICE_ERROR
LUNA_RET_RETRY	0x00300021	CKR_DEVICE_ERROR
LUNA_RET_SHA1_RSA_SELF_TEST_FAILURE	0x00300022	CKR_DEVICE_ERROR
LUNA_RET_SELF_TEST_FAILURE	0x00300023	CKR_DEVICE_ERROR
LUNA_RET_INCOMPATIBLE	0x00300024	CKR_DEVICE_ERROR
LUNA_RET_RIPEMD160_SELF_TEST_FAILURE	0x00300034	CKR_DEVICE_ERROR
LUNA_RET_TOKEN_LOCKED_OUT_CL	0x00300100	CKR_DEVICE_ERROR
LUNA_RET_TOKEN_LOCKED_OUT_MM	0x00300101	CKR_DEVICE_ERROR
LUNA_RET_TOKEN_LOCKED_OUT_UM	0x00300102	CKR_DEVICE_ERROR
LUNA_RET_TOKEN_LOCKED_OUT_SM	0x00300103	CKR_DEVICE_ERROR
LUNA_RET_TOKEN_LOCKED_OUT_RN	0x00300104	CKR_DEVICE_ERROR
LUNA_RET_TOKEN_LOCKED_OUT_CA	0x00300105	CKR_DEVICE_ERROR
LUNA_RET_TOKEN_LOCKED_OUT_PM	0x00300106	CKR_DEVICE_ERROR
LUNA_RET_TOKEN_LOCKED_OUT_OH	0x00300107	CKR_DEVICE_ERROR
LUNA_RET_TOKEN_LOCKED_OUT_CCM	0x00300108	CKR_DEVICE_ERROR
LUNA_RET_TOKEN_LOCKED_OUT_SHA_DIGEST	0x00300109	CKR_DEVICE_ERROR
LUNA_RET_SM_ACCESS_REALLOC_ERROR	0x00310101	CKR_DEVICE_ERROR

HSM Error	Hex Code	PKCS#11 or SFNT Defined CKR Error
LUNA_RET_SM_SESSION_REALLOC_ERROR	0x00310102	CKR_DEVICE_ERROR
LUNA_RET_SM_MEMORY_ALLOCATION_ERROR	0x00310103	CKR_DEVICE_ERROR
LUNA_RET_ENCRYPTED_DATA_INVALID	0x00400000	CKR_ENCRYPTED_DATA_INVALID
LUNA_RET_ENCRYPTED_DATA_LEN_RANGE	0x00410000	CKR_ENCRYPTED_DATA_LEN_ RANGE
LUNA_RET_FUNCTION_CANCELED	0x00500000	CKR_FUNCTION_CANCELED
LUNA_RET_KEY_HANDLE_INVALID	0x00600000	CKR_KEY_HANDLE_INVALID
LUNA_RET_MECHANISM_INVALID	0x00700000	CKR_MECHANISM_INVALID
LUNA_RET_MECHANISM_PARAM_INVALID	0x00710000	CKR_MECHANISM_PARAM_ INVALID
LUNA_RET_OPERATION_ACTIVE	0x00900000	CKR_OPERATION_ACTIVE
LUNA_RET_OPERATION_NOT_INITIALIZED	0x00910000	CKR_OPERATION_NOT_ INITIALIZED
LUNA_RET_UM_PIN_INCORRECT	0x00a00000	CKR_PIN_INCORRECT
LUNA_RET_UM_PIN_INCORRECT_CONTAINER_ ZEROIZED	0x00a00001	CKR_PIN_INCORRECT
LUNA_RET_UM_PIN_INCORRECT_CONTAINER_ LOCKED	0x00a00002	CKR_PIN_INCORRECT
LUNA_RET_UM_PIN_LEN_RANGE	0x00a20000	CKR_PIN_LEN_RANGE
LUNA_RET_SM_PIN_EXPIRED	0x00a30000	CKR_PIN_EXPIRED
LUNA_RET_SM_EXCLUSIVE_SESSION_EXISTS	0x00b20000	CKR_SESSION_EXCLUSIVE_ EXISTS
LUNA_RET_SM_SESSION_HANDLE_INVALID	0x00b30000	CKR_SESSION_HANDLE_INVALID
LUNA_RET_SIGNATURE_INVALID	0x00c00000	CKR_SIGNATURE_INVALID
LUNA_RET_SIGNATURE_LEN_RANGE	0x00c10000	CKR_SIGNATURE_LEN_RANGE

HSM Error	Hex Code	PKCS#11 or SFNT Defined CKR Error
LUNA_RET_UNWRAPPING_KEY_HANDLE_INVALID	0x00f00000	CKR_UNWRAPPING_KEY_HANDLE_INVALID
LUNA_RET_UNWRAPPING_KEY_SIZE_RANGE	0x00f10000	CKR_UNWRAPPING_KEY_SIZE_RANGE
LUNA_RET_UNWRAPPING_KEY_TYPE_INCONSISTENT	0x00f20000	CKR_UNWRAPPING_KEY_TYPE_INCONSISTENT
LUNA_RET_USER_ALREADY_LOGGED_IN	0x01000000	CKR_USER_ALREADY_LOGGED_IN
LUNA_RET_SM_OTHER_USER_LOGGED_IN	0x01000001	CKR_USER_ALREADY_LOGGED_IN
LUNA_RET_USER_NOT_LOGGED_IN	0x01010000	CKR_USER_NOT_LOGGED_IN
LUNA_RET_SM_NOT_LOGGED_IN	0x01010001	CKR_USER_NOT_LOGGED_IN
LUNA_RET_USER_PIN_NOT_INITIALIZED	0x01020000	CKR_USER_PIN_NOT_INITIALIZED
LUNA_RET_USER_TYPE_INVALID	0x01030000	CKR_USER_TYPE_INVALID
LUNA_RET_WRAPPED_KEY_INVALID	0x01100000	CKR_WRAPPED_KEY_INVALID
LUNA_RET_WRAPPED_KEY_LEN_RANGE	0x01120000	CKR_WRAPPED_KEY_LEN_RANGE
LUNA_RET_WRAPPING_KEY_HANDLE_INVALID	0x01130000	CKR_WRAPPING_KEY_HANDLE_INVALID
LUNA_RET_WRAPPING_KEY_SIZE_RANGE	0x01140000	CKR_WRAPPING_KEY_SIZE_RANGE
LUNA_RET_WRAPPING_KEY_TYPE_INCONSISTENT	0x01150000	CKR_WRAPPING_KEY_TYPE_INCONSISTENT
LUNA_RET_CERT_VERSION_NOT_SUPPORTED	0x00300300	CKR_DEVICE_ERROR
LUNA_RET_SIM_AUTHFORM_INVALID	0x0020011e	CKR_SIM_AUTHFORM_INVALID
LUNA_RET_CCM_TOO_LARGE	0x00210001	CKR_DATA_LEN_RANGE
LUNA_RET_TEST_VS_BSAFE_FAILED	0x00300820	CKR_DEVICE_ERROR

HSM Error	Hex Code	PKCS#11 or SFNT Defined CKR Error
LUNA_RET_SFNT3120_ERROR	0x00300821	CKR_DEVICE_ERROR
LUNA_RET_SFNT3120_SELFTEST_FAILED	0x00300822	CKR_DEVICE_ERROR
LUNA_RET_SFNT3120_CRC	0x00300823	CKR_DEVICE_ERROR
LUNA_RET_SFNT3120_ALG_NO_SOFTWARE_ SUPPORT	0x00300824	CKR_DEVICE_ERROR
LUNA_RET_ISES_ERROR	0x00300880	CKR_DEVICE_ERROR
LUNA_RET_ISES_INIT_FAILED	0x00300881	CKR_DEVICE_ERROR
LUNA_RET_ISES_LNAU_TEST_FAILED	0x00300882	CKR_DEVICE_ERROR
LUNA_RET_ISES_RNG_TEST_FAILED	0x00300883	CKR_DEVICE_ERROR
LUNA_RET_ISES_CMD_FAILED	0x00300884	CKR_DEVICE_ERROR
LUNA_RET_ISES_CMD_PARAMETER_INVALID	0x00300885	CKR_DEVICE_ERROR
LUNA_RET_ISES_TEST_VS_BSAFE_FAILED	0x00300886	CKR_DEVICE_ERROR
LUNA_RET_RM_ELEMENT_VALUE_INVALID	0x00200a00	CKR_DATA_INVALID
LUNA_RET_RM_ELEMENT_ID_INVALID	0x00200a01	CKR_DATA_INVALID
LUNA_RET_RM_NO_MEMORY	0x00310a02	CKR_DEVICE_MEMORY
LUNA_RET_RM_BAD_HSM_PARAMS	0x00300a03	CKR_DEVICE_ERROR
LUNA_RET_RM_POLICY_ELEMENT_ DESTRUCTIVE	0x00200a04	CKR_DATA_INVALID
LUNA_RET_RM_POLICY_ELEMENT_NOT_ DESTRUCTIVE	0x00200a05	CKR_DATA_INVALID
LUNA_RET_RM_CONFIG_CHANGE_ILLEGAL	0x00010a06	CKR_CANCEL
LUNA_RET_RM_CONFIG_CHANGE_FAILS_ DEPENDENCIES	0x00010a07	CKR_CANCEL
LUNA_RET_LICENSE_ID_UNKNOWN	0x00200a08	CKR_DATA_INVALID

HSM Error	Hex Code	PKCS#11 or SFNT Defined CKR Error
LUNA_RET_LICENSE_CAPACITY_EXCEEDED	0x00010a09	CKR_LICENSE_CAPACITY_ EXCEEDED
LUNA_RET_RM_POLICY_WRITE_RESTRICTED	0x00010a0a	CKR_CANCEL
LUNA_RET_OPERATION_RESTRICTED	0x00010a0b	CKR_OPERATION_NOT_ALLOWED
LUNA_RET_CANNOT_PERFORM_OPERATION_ TWICE	0x00010a0c	CKR_CANCEL
LUNA_RET_BAD_PPID	0x00200a0d	CKR_DATA_INVALID
LUNA_RET_BAD_FW_VERSION	0x00200a0e	CKR_DATA_INVALID
LUNA_RET_OPERATION_SHOULD_BE_ DESTRUCTIVE	0x00200a0f	CKR_DATA_INVALID
LUNA_RET_RM_CONFIG_ILLEGAL	0x00200a10	CKR_DATA_INVALID
LUNA_RET_BAD_SN	0x00200a11	CKR_DATA_INVALID
LUNA_RET_CHALLENGE_TYPE_INVALID	0x00200b00	CKR_DATA_INVALID
LUNA_RET_CHALLENGE_REQUIRES_PED	0x00010b01	CKR_CANCEL
LUNA_RET_CHALLENGE_NOT_REQUIRED	0x00010b02	CKR_CANCEL
LUNA_RET_CHALLENGE_RESPONSE_ INCORRECT	0x00a00b03	CKR_PIN_INCORRECT
LUNA_RET_OH_OBJECT_VERSION_INVALID	0x00300c00	CKR_DEVICE_ERROR
LUNA_RET_OH_OBJECT_TYPE_INVALID	0x00300c01	CKR_DEVICE_ERROR
LUNA_RET_OH_OBJECT_ALREADY_EXISTS	0x00010c02	CKR_CANCEL
LUNA_RET_OH_OBJECT_OWNER_DOES_NOT_ EXIST	0x00200c03	CKR_DATA_INVALID
LUNA_RET_STORAGE_TYPE_INCONSISTENT	0x00200c04	CKR_DATA_INVALID
LUNA_RET_CONTAINER_CAN_NOT_HAVE_ MEMBERS	0x00200c05	CKR_DATA_INVALID

HSM Error	Hex Code	PKCS#11 or SFNT Defined CKR Error
LUNA_RET_SAVED_STATE_INVALID	0x01600000	CKR_SAVED_STATE_INVALID
LUNA_RET_STATE_UNSAVEABLE	0x01800000	CKR_STATE_UNSAVEABLE
LUNA_RET_ERROR	0x80000000	CKR_GENERAL_ERROR
LUNA_RET_CONTAINER_HANDLE_INVALID	0x80000001	CKR_CONTAINER_HANDLE_ INVALID
LUNA_RET_INVALID_PADDING_TYPE	0x80000002	CKR_DATA_INVALID
LUNA_RET_NOT_FOUND	0x80000007	CKR_FUNCTION_FAILED
LUNA_RET_TOO_MANY_CONTAINERS	0x80000008	CKR_TOO_MANY_CONTAINERS
LUNA_RET_CONTAINER_LOCKED	0x80000009	CKR_PIN_LOCKED
LUNA_RET_CONTAINER_IS_DISABLED	0x8000000a	CKR_PARTITION_DISABLED
LUNA_RET_SECURITY_PARAMETER_MISSING	0x8000000b	CKR_SECURITY_PARAMETER_ MISSING
LUNA_RET_DEVICE_TIMEOUT	0x8000000c	CKR_DEVICE_TIMEOUT
LUNA_RET_OBJECT_DELETED	0x8000000d	HSM Internal ONLY
LUNA_RET_INVALID_FUF_TARGET	0x8000000e	CKR_INVALID_FUF_TARGET
LUNA_RET_INVALID_FUF_HEADER	0x8000000f	CKR_INVALID_FUF_HEADER
LUNA_RET_INVALID_FUF_VERSION	0x80000010	CKR_INVALID_FUF_VERSION
LUNA_RET_KCV_PARAMETER_ALREADY_EXISTS	0x80000100	CKR_CLONING_PARAMETER_ ALREADY_EXISTS
LUNA_RET_KCV_PARAMETER_COULD_NOT_BE_ ADDED	0x80000101	CKR_DEVICE_MEMORY
LUNA_RET_INVALID_CERTIFICATE_DATA	0x80000102	CKR_CERTIFICATE_DATA_ INVALID
LUNA_RET_INVALID_CERTIFICATE_TYPE	0x80000103	CKR_CERTIFICATE_DATA_ INVALID

HSM Error	Hex Code	PKCS#11 or SFNT Defined CKR Error
LUNA_RET_INVALID_CERTIFICATE_VERSION	0x80000104	CKR_CERTIFICATE_DATA_ INVALID
LUNA_RET_INVALID_MODULUS_SIZE	0x80000105	CKR_ATTRIBUTE_VALUE_INVALID
LUNA_RET_WRAPPING_ERROR	0x80000107	CKR_WRAPPING_ERROR
LUNA_RET_UNWRAPPING_ERROR	0x80000108	CKR_UNWRAPPING_ERROR
LUNA_RET_INVALID_PRIVATE_KEY_TYPE	0x80000109	CKR_DATA_INVALID
LUNA_RET_TSN_MISMATCH	0x8000010a	CKR_DATA_INVALID
LUNA_RET_KCV_PARAMETER_MISSING	0x8000010b	CKR_CLONING_PARAMETER_ MISSING
LUNA_RET_TWC_PARAMETER_MISSING	0x8000010c	CKR_CERTIFICATE_DATA_ MISSING
LUNA_RET_TUK_PARAMETER_MISSING	0x8000010d	CKR_CERTIFICATE_DATA_ MISSING
LUNA_RET_CPK_PARAMETER_MISSING	0x8000010e	CKR_KEY_NEEDED
LUNA_RET_MASKING_NOT_SUPPORTED	0x8000010f	CKR_FUNCTION_NOT_ SUPPORTED
LUNA_RET_INVALID_ACCESS_LEVEL	0x80000110	CKR_ARGUMENTS_BAD
LUNA_RET_MAC_MISSING	0x80000111	CKR_MAC_MISSING
LUNA_RET_DAC_POLICY_PID_MISMATCH	0x80000112	CKR_DAC_POLICY_PID_ MISMATCH
LUNA_RET_DAC_MISSING	0x80000113	CKR_DAC_MISSING
LUNA_RET_BAD_DAC	0x80000114	CKR_BAD_DAC
LUNA_RET_SSK_MISSING	0x80000115	CKR_SSK_MISSING
LUNA_RET_BAD_MAC	0x80000116	CKR_BAD_MAC
LUNA_RET_DAK_MISSING	0x80000117	CKR_DAK_MISSING

HSM Error	Hex Code	PKCS#11 or SFNT Defined CKR Error
LUNA_RET_BAD_DAK	0x80000118	CKR_BAD_DAK
LUNA_RET_HOK_MISSING	0x80000119	CKR_CERTIFICATE_DATA_ MISSING
LUNA_RET_CITS_DAK_MISSING	0x8000011a	CKR_CITS_DAK_MISSING
LUNA_RET_SIM_AUTHORIZATION_FAILED	0x8000011b	CKR_SIM_AUTHORIZATION_ FAILED
LUNA_RET_SIM_VERSION_UNSUPPORTED	0x8000011c	CKR_SIM_VERSION_ UNSUPPORTED
LUNA_RET_SIM_CORRUPT_DATA	0x8000011d	CKR_SIM_CORRUPT_DATA
LUNA_RET_ECC_MIC_MISSING	0x8000011e	CKR_CERTIFICATE_DATA_ MISSING
LUNA_RET_ECC_HOK_MISSING	0x8000011f	CKR_CERTIFICATE_DATA_ MISSING
LUNA_RET_ECC_HOC_MISSING	0x80000120	CKR_CERTIFICATE_DATA_ MISSING
LUNA_RET_ECC_DAK_MISSING	0x80000121	CKR_CERTIFICATE_DATA_ MISSING
LUNA_RET_ECC_DAC_MISSING	0x80000122	CKR_CERTIFICATE_DATA_ MISSING
LUNA_RET_ROOT_CERT_MISSING	0x80000123	CKR_CERTIFICATE_DATA_ MISSING
LUNA_RET_HOC_MISSING	0x80000124	CKR_CERTIFICATE_DATA_ MISSING
LUNA_RET_INVALID_CERTIFICATE_FUNCTION	0x80000125	CKR_CERTIFICATE_DATA_ INVALID
LUNA_RET_N_TOO_LARGE	0x80000200	CKR_ARGUMENTS_BAD
LUNA_RET_N_TOO_SMALL	0x80000201	CKR_ARGUMENTS_BAD
LUNA_RET_M_TOO_LARGE	0x80000202	CKR_ARGUMENTS_BAD

HSM Error	Hex Code	PKCS#11 or SFNT Defined CKR Error
LUNA_RET_M_TOO_SMALL	0x80000203	CKR_ARGUMENTS_BAD
LUNA_RET_WEIGHT_TOO_LARGE	0x80000204	CKR_ARGUMENTS_BAD
LUNA_RET_WEIGHT_TOO_SMALL	0x80000205	CKR_ARGUMENTS_BAD
LUNA_RET_TOTAL_WEIGHT_INVALID	0x80000206	CKR_ARGUMENTS_BAD
LUNA_RET_MISSING_SPLITS	0x80000207	CKR_ARGUMENTS_BAD
LUNA_RET_SPLIT_DATA_INVALID	0x80000208	CKR_ARGUMENTS_BAD
LUNA_RET_SPLIT_ID_INVALID	0x80000209	CKR_ARGUMENTS_BAD
LUNA_RET_M_OF_N_PARAMETER_NOT_ AVAILABLE	0x8000020a	CKR_OPERATION_NOT_ INITIALIZED
LUNA_RET_M_OF_N_ACTIVATION_REQUIRED	0x8000020b	CKR_OPERATION_NOT_ INITIALIZED
LUNA_RET_TOO_MANY_WEIGHTS	0x8000020e	CKR_ARGUMENTS_BAD
LUNA_RET_MISSING_WEIGHT_VALUE	0x8000020f	CKR_ARGUMENTS_BAD
LUNA_RET_MISSING_VALUE_FOR_M	0x80000210	CKR_ARGUMENTS_BAD
LUNA_RET_MISSING_VALUE_FOR_N	0x80000211	CKR_ARGUMENTS_BAD
LUNA_RET_MISSING_NUMBER_OF_VECTORS	0x80000212	CKR_ARGUMENTS_BAD
LUNA_RET_MISSING_VECTOR	0x80000213	CKR_ARGUMENTS_BAD
LUNA_RET_VECTOR_TOO_LARGE	0x80000214	CKR_ARGUMENTS_BAD
LUNA_RET_VECTOR_TOO_SMALL	0x80000215	CKR_ARGUMENTS_BAD
LUNA_RET_TOO_MANY_VECTORS_PROVIDED	0x80000216	CKR_ARGUMENTS_BAD
LUNA_RET_INVALID_VECTOR_SIZE	0x80000217	CKR_ARGUMENTS_BAD
LUNA_RET_M_OF_N_PARAMETER_EXIST	0x80000218	CKR_FUNCTION_FAILED
LUNA_RET_VECTOR_VERSION_INVALID	0x80000219	CKR_DATA_INVALID

HSM Error	Hex Code	PKCS#11 or SFNT Defined CKR Error
LUNA_RET_VECTOR_OF_DIFFERENT_SET	0x8000021a	CKR_ARGUMENTS_BAD
LUNA_RET_VECTOR_DUPLICATE	0x8000021b	CKR_ARGUMENTS_BAD
LUNA_RET_VECTOR_TYPE_INVALID	0x8000021c	CKR_ARGUMENTS_BAD
LUNA_RET_MISSING_COMMAND_PARAMETER	0x8000021d	CKR_ARGUMENTS_BAD
LUNA_RET_M_OF_N_CLONING_IS_NOT_ ALLOWED	0x8000021e	CKR_FUNCTION_NOT_ SUPPORTED
LUNA_RET_M_OF_N_IS_NOT_REQUIRED	0x8000021f	CKR_OPERATION_NOT_ INITIALIZED
LUNA_RET_M_OF_N_IS_NOT_INITIALZED	0x80000220	CKR_OPERATION_NOT_ INITIALIZED
LUNA_RET_M_OF_N_SECRET_INVALID	0x80000221	CKR_GENERAL_ERROR
LUNA_RET_CCM_NOT_PRESENT	0x80000300	CKR_FUNCTION_NOT_ SUPPORTED
LUNA_RET_CCM_NOT_SUPPORTED	0x80000301	CKR_FUNCTION_NOT_ SUPPORTED
LUNA_RET_CCM_UNREMOVABLE	0x80000302	CKR_DATA_INVALID
LUNA_RET_CCM_CERT_INVALID	0x80000303	CKR_DATA_INVALID
LUNA_RET_CCM_SIGN_INVALID	0x80000304	CKR_DATA_INVALID
LUNA_RET_CCM_UPDATE_DENIED	0x80000305	CKR_DATA_INVALID
LUNA_RET_CCM_FWUPDATE_DENIED	0x80000306	CKR_DATA_INVALID
LUNA_RET_SM_ACCESS_ID_INVALID	0x80000400	CKR_DATA_INVALID
LUNA_RET_SM_ACCESS_ALREADY_EXISTS	0x80000401	CKR_DATA_INVALID
LUNA_RET_SM_MULTIPLE_ACCESS_DISABLED	0x80000402	CKR_FUNCTION_NOT_ SUPPORTED
LUNA_RET_SM_UNKNOWN_ACCESS_TYPE	0x80000403	CKR_ARGUMENTS_BAD

HSM Error	Hex Code	PKCS#11 or SFNT Defined CKR Error
LUNA_RET_SM_BAD_ACCESS_HANDLE	0x80000404	CKR_DATA_INVALID
LUNA_RET_SM_BAD_CONTEXT_NUMBER	0x80000405	CKR_DATA_INVALID
LUNA_RET_SM_UNKNOWN_SESSION_TYPE	0x80000406	CKR_DATA_INVALID
LUNA_RET_SM_CONTEXT_ALREADY_ ALLOCATED	0x80000407	CKR_DATA_INVALID
LUNA_RET_SM_CONTEXT_NOT_ALLOCATED	0x80000408	CKR_DEVICE_MEMORY
LUNA_RET_SM_CONTEXT_BUFFER_OVERFLOW	0x80000409	CKR_DEVICE_MEMORY
LUNA_RET_SM_TOSM_DOES_NOT_VALIDATE	0x8000040A	CKR_USER_NOT_LOGGED_IN
LUNA_RET_SM_ACCESS_DOES_NOT_VALIDATE	0x8000040B	CKR_USER_NOT_AUTHORIZED
LUNA_RET_MTK_ZEROIZED	0x80000531	CKR_MTK_ZEROIZED
LUNA_RET_MTK_STATE_INVALID	0x80000532	CKR_MTK_STATE_INVALID
LUNA_RET_MTK_SPLIT_INVALID	0x80000533	CKR_MTK_SPLIT_INVALID
LUNA_RET_INVALID_IP_PACKET	0x80000600	CKR_DEVICE_ERROR
LUNA_RET_INVALID_BOARD_TYPE	0x80000700	CKR_DEVICE_ERROR
LUNA_RET_ECC_NOT_SUPPORTED	0x80000601	CKR_FUNCTION_NOT_ SUPPORTED
LUNA_RET_ECC_BUFFER_OVERFLOW	0x80000602	CKR_DEVICE_ERROR
LUNA_RET_ECC_POINT_INVALID	0x80000603	CKR_ECC_POINT_INVALID**
LUNA_RET_ECC_SELF_TEST_FAILURE	0x80000604	CKR_DEVICE_ERROR
LUNA_RET_ECC_UNKNOWN_CURVE	0x80000605	CKR_ECC_UNKNOWN_CURVE
LUNA_RET_HA_NOT_SUPPORTED	0x80000900	CKR_FUNCTION_NOT_ SUPPORTED
LUNA_RET_HA_USER_NOT_INITIALIZED	0x80000901	CKR_OPERATION_NOT_ INITIALIZED

HSM Error	Hex Code	PKCS#11 or SFNT Defined CKR Error
LUNA_RET_HSM_STORAGE_FULL	0x80000902	CKR_HSM_STORAGE_FULL
LUNA_RET_CONTAINER_OBJECT_STORAGE_FULL	0x80000903	CKR_CONTAINER_OBJECT_STORAGE_FULL
LUNA_RET_KEY_NOT_ACTIVE	0x80000904	CKR_KEY_NOT_ACTIVE
LUNA_RET_CB_NOT_SUPPORTED	0x80000a01	CKR_FUNCTION_NOT_SUPPORTED
LUNA_RET_CB_PARAM_INVALID	0x80000a02	CKR_CALLBACK_ERROR
LUNA_RET_CB_NO_MEMORY	0x80000a03	CKR_DEVICE_MEMORY
LUNA_RET_CB_TIMEOUT	0x80000a04	CKR_CALLBACK_ERROR
LUNA_RET_CB_RETRY	0x80000a05	CKR_CALLBACK_ERROR
LUNA_RET_CB_ABORTED	0x80000a06	CKR_CALLBACK_ERROR
LUNA_RET_CB_SYS_ERROR	0x80000a07	CKR_CALLBACK_ERROR
LUNA_RET_CB_HIOS_HANDLE_INVALID	0x80000a10	CKR_CALLBACK_ERROR
LUNA_RET_CB_HIOS_ID_INVALID	0x80000a11	CKR_CALLBACK_ERROR
LUNA_RET_CB_HIOS_CLOSED	0x80000a12	CKR_CALLBACK_ERROR
LUNA_RET_CB_HIOS_CANCELED	0x80000a13	CKR_CALLBACK_ERROR
LUNA_RET_CB_HIOS_IO_ERROR	0x80000a14	CKR_CALLBACK_ERROR
LUNA_RET_CB_HIOS_SEND_TIMEOUT	0x80000a15	CKR_CALLBACK_ERROR
LUNA_RET_CB_HIOS_RECV_TIMEOUT	0x80000a16	CKR_CALLBACK_ERROR
LUNA_RET_CB_HIOS_STATE_INVALID	0x80000a17	CKR_CALLBACK_ERROR
LUNA_RET_CB_HIOS_OUTPUT_BUFFER_TOO_SMALL	0x80000a18	CKR_CALLBACK_ERROR
LUNA_RET_CB_HIOS_INPUT_BUFFER_TOO_SMALL	0x80000a19	CKR_CALLBACK_ERROR

HSM Error	Hex Code	PKCS#11 or SFNT Defined CKR Error
LUNA_RET_CB_HANDLE_INVALID	0x80000a20	CKR_CALLBACK_ERROR
LUNA_RET_CB_ID_INVALID	0x80000a21	CKR_CALLBACK_ERROR
LUNA_RET_CB_REMOTE_ABORT	0x80000a22	CKR_CALLBACK_ERROR
LUNA_RET_CB_REMOTE_CLOSED	0x80000a23	CKR_CALLBACK_ERROR
LUNA_RET_CB_REMOTE_ABANDONED	0x80000a24	CKR_CALLBACK_ERROR
LUNA_RET_CB_MUST_READ	0x80000a25	CKR_CALLBACK_ERROR
LUNA_RET_CB_MUST_WRITE	0x80000a26	CKR_CALLBACK_ERROR
LUNA_RET_CB_INVALID_CALL_FOR_THE_STATE	0x80000a27	CKR_CALLBACK_ERROR
LUNA_RET_CB_SYNC_ERROR	0x80000a28	CKR_CALLBACK_ERROR
LUNA_RET_CB_PROT_DATA_INVALID	0x80000a29	CKR_CALLBACK_ERROR
LUNA_RET_LOG_FILE_NOT_OPEN	0x80000d00	CKR_LOG_FILE_NOT_OPEN
LUNA_RET_LOG_FILE_WRITE_ERROR	0x80000d01	CKR_LOG_FILE_WRITE_ERROR
LUNA_RET_LOG_BAD_FILE_NAME	0x80000d02	CKR_LOG_BAD_FILE_NAME
LUNA_RET_LOG_FULL	0x80000d03	CKR_LOG_FULL
LUNA_RET_LOG_NO_KCV	0x80000d04	CKR_LOG_NO_KCV
LUNA_RET_LOG_BAD_RECORD_HMAC	0x80000d05	CKR_LOG_BAD_RECORD_HMAC
LUNA_RET_LOG_BAD_TIME	0x80000d06	CKR_LOG_BAD_TIME
LUNA_RET_LOG_AUDIT_NOT_INITIALIZED	0x80000d07	CKR_LOG_AUDIT_NOT_ INITIALIZED
LUNA_RET_LOG_RESYNC_NEEDED	0x80000d08	CKR_LOG_RESYNC_NEEDED
LUNA_RET_AUDIT_LOGIN_TIMEOUT_IN_ PROGRESS	0x80000d09	CKR_AUDIT_LOGIN_TIMEOUT_IN_ PROGRESS
LUNA_RET_AUDIT_LOGIN_FAILURE_THRESHOLD	0x80000d0a	CKR_AUDIT_LOGIN_FAILURE_ THRESHOLD

* This error (CKR_TEMPLATE_INCONSISTENT) might be encountered when using CKDemo in a new client with firmware older than version 6.22.0. Try CKDemo option 98, sub-option 16. If it is set to "enhanced roles", try selecting it to set it to "legacy Luna roles". The setting is a toggle, and flips every time you select it.

** This error, or "unable to read public key", might be encountered when using BSAFE to encrypt data with ECC public key using curves from the Brainpool suite. As indicated on the BSAFE website (May 2012) they do not appear to support Brainpool curves. Therefore, your own applications should not attempt that combination, and you should avoid attempting to specify Brainpool curves with BSAFE ECC when using SafeNet's CKDemo utility.

Library Codes

Hex value	Decimal value	Return code/error description
0	0	OKAY, NO ERROR
0xC0000000	3221225472	PROGRAMMING ERROR: RETURN CODE
0xC0000001	3221225473	OUT OF MEMORY
0xC0000002	3221225474	NON-SPECIFIC ERROR
0xC0000003	3221225475	UNEXPECTED NULL POINTER
0xC0000004	3221225476	PROGRAMMING ERROR: LOGIC
0xC0000005	3221225477	OPERATION WOULD BLOCK IF ATTEMPTED
0xC0000006	3221225478	BUFFER IS TOO SMALL
0xC0000100	3221225728	OPERATION CANCEL
0xC0000101	3221225729	INVALID SLOT IDENTIFIER
0xC0000102	3221225730	INVALID DATA
0xC0000103	3221225731	INVALID PIN
0xC0000104	3221225732	NO TOKEN PRESENT
0xC0000105	3221225733	FUNCTION IS NOT SUPPORTED
0xC0000106	3221225734	NON-CRYPTOKI ELEMENT CLONE
0xC0000107	3221225735	INVALID BUFFER SIZE FOR CHALLENGE

Hex value	Decimal value	Return code/error description
0xC0000108	3221225736	PIN IS LOCKED
0xC0000109	3221225737	INVALID VERSION
0xC000010a	3221225738	NEEDED KEY NOT PROVIDED
0xC000010b	3221225739	USER NAME IS IN USE
0xC0000200	3221225984	INVALID DISTINGUISHED ENCODING RULES CLASS
0xC0000303	3221226243	OPERATION TIMED OUT
0xC0000304	3221226244	RESET FAILED
0xC0000400	3221226496	INVALID TOKEN STATE
0xC0000401	3221226497	DATA APPEARS CORRUPTED
0xC0000402	3221226498	INVALID FILENAME
0xC0000403	3221226499	FILE IS READ-ONLY
0xC0000404	3221226500	FILE ERROR
0xC0000405	3221226501	INVALID OBJECT IDENTIFIER
0xC0000406	3221226502	INVALID SOCKET ADDRESS
0xC0000407	3221226503	INVALID LISTEN SOCKET
0xC0000408	3221226504	CACHE IS NOT CURRENT
0xC0000409	3221226505	CACHE IS NOT MAPPED
0xC000040a	3221226506	OBJECT IS NOT IN LIST
0xC000040b	3221226507	INVALID INDEX
0xC000040c	3221226508	OBJECT ALREADY EXISTS
0xC000040d	3221226509	SEMAPHORE ERROR
0xC000040e	3221226510	END OF LIST ENCOUNTERED

Hex value	Decimal value	Return code/error description
0xC000040f	3221226511	WOULD ASSIGN SAME VALUE
0xC0000410	3221226512	INVALID GROUP NAME
0xC0000411	3221226513	NOT HSM BACKUP TOKEN
0xC0000412	3221226514	NOT PARTITION BACKUP TOKEN
0xC0000413	3221226515	SIM NOT SUPPORTED
0xC0000500	3221226752	SOCKET ERROR
0xC0000501	3221226753	SOCKET WRITE ERROR
0xC0000502	3221226754	SOCKET READ ERROR
0xC0000503	3221226755	CLIENT MESSAGE ERROR
0xC0000504	3221226756	SERVER DISCONNECTED
0xC0000505	3221226757	CLIENT DISCONNECTED
0xC0000506	3221226758	SOCKET WOULD BLOCK
0xC0000507	3221226759	SOCKET ADDRESS IS IN USE
0xC0000508	3221226760	SOCKET BAD FILE DESCRIPTOR
0xC0000509	3221226761	HOST RESOLUTION ERROR
0xC000050a	3221226762	INVALID HOST CERTIFICATE
0xC0000600	3221227008	NO BUFFER AVAILABLE
0xC0000601	3221227009	INVALID ENUMERATION OPTION
0xC0000700	3221227264	SSL ERROR
0xC0000701	3221227265	SSL CTX ERROR
0xC0000702	3221227266	SSL CIPHER LIST ERROR
0xC0000703	3221227267	SSL CERT VERIFICATION LOCATION ERROR

Hex value	Decimal value	Return code/error description
0xC0000704	3221227268	SSL LOAD SERVER CERT ERROR
0xC0000705	3221227269	SSL LOAD SERVER PRIVATE KEY ERROR
0xC0000706	3221227270	SSL VALIDATE SERVER PRIVATE KEY ERROR
0xC0000707	3221227271	SSL CREATE SSL ERROR
0xC0000708	3221227272	SSL LOAD CLIENT CERT ERROR
0xC0000709	3221227273	SSL GET CERTIFICATE ERROR
0xC000070a	3221227274	SSL INVALID CERT STRUCTURE
0xC000070b	3221227275	SSL LOAD CLIENT PRIVATE KEY ERROR
0xC000070c	3221227276	SSL GET PEER CERT ERROR
0xC000070d	3221227277	SSL WANT READ ERROR
0xC000070e	3221227278	SSL WANT WRITE ERROR
0xC000070f	3221227279	SSL WANT X509 LOOKUP ERROR
0xC0000710	3221227280	SSL SYSCALL ERROR
0xC0000711	3221227281	SSL FAILED HANDSHAKE
0xC0000800	3221227520	INVALID CERTIFICATE TYPE
0xC0000900	3221227776	INVALID PORT
0xC0000901	3221227777	SESSION SCRIPT EXISTS
0xC0001000	3221229568	PARTITION LOCKED
0xC0001001	3221229569	PARTITION NOT ACTIVATED
0xc0002000	3221233664	FAILED TO CREATE THREAD
0xc0002001	3221233665	CALLBACK ERROR
0xc0002002	3221233666	UNKNOWN CALLBACK COMMAND

Hex value	Decimal value	Return code/error description
0xc0002003	3221233667	SHUTTING DOWN
0xc0002004	3221233668	REMOTE SIDE DISCONNECTED
0xc0002005	3221233669	SOCKET CLOSED
0xC0002006	3221233670	INVALID COMMAND
0xC0002007	3221233671	UNKNOWN COMMAND
0xC0002008	3221233672	UNKNOWN COMMAND VERSION
0xC0002009	3221233673	FILE LOCK FAILED
0xC0002010	3221233680	FILE LOCK ERROR
0xc0002011	3221233681	FAILED TO CREATE PROCESS
0xc0002012	3221233682	USB PED NOT FOUND
0xc0002013	3221233683	USB PED NOT RESPONDING
0xc0002014	3221233684	USB PED OPERATION CANCELLED
0xc0002015	3221233685	USB PED TOO MANY CONNECTED
0xc0002016	3221233686	USB PED OUT OF SYNC
0xC0001100	3221229824	UNABLE TO CONNECT

Vendor-Defined Return Codes

Code	Name
0x00000141	CKR_INSERTION_CALLBACK_NOT_SUPPORTED
0x0052	CKR_FUNCTION_PARALLEL
0x00B2	CKR_SESSION_EXCLUSIVE_EXISTS
(CKR_VENDOR_DEFINED + 0x04)	CKR_RC_ERROR
(CKR_VENDOR_DEFINED + 0x05)	CKR_CONTAINER_HANDLE_INVALID

Code	Name
(CKR_VENDOR_DEFINED + 0x06)	CKR_TOO_MANY_CONTAINERS
(CKR_VENDOR_DEFINED + 0x07)	CKR_USER_LOCKED_OUT
(CKR_VENDOR_DEFINED + 0x08)	CKR_CLONING_PARAMETER_ALREADY_EXISTS
(CKR_VENDOR_DEFINED + 0x09)	CKR_CLONING_PARAMETER_MISSING
(CKR_VENDOR_DEFINED + 0x0a)	CKR_CERTIFICATE_DATA_MISSING
(CKR_VENDOR_DEFINED + 0x0b)	CKR_CERTIFICATE_DATA_INVALID
(CKR_VENDOR_DEFINED + 0x0c)	CKR_ACCEL_DEVICE_ERROR
(CKR_VENDOR_DEFINED + 0x0d)	CKR_WRAPPING_ERROR
(CKR_VENDOR_DEFINED + 0x0e)	CKR_UNWRAPPING_ERROR
(CKR_VENDOR_DEFINED + 0x0f)	CKR_MAC_MISSING
(CKR_VENDOR_DEFINED + 0x10)	CKR_DAC_POLICY_PID_MISMATCH
(CKR_VENDOR_DEFINED + 0x11)	CKR_DAC_MISSING
(CKR_VENDOR_DEFINED + 0x12)	CKR_BAD_DAC
(CKR_VENDOR_DEFINED + 0x13)	CKR_SSK_MISSING
(CKR_VENDOR_DEFINED + 0x14)	CKR_BAD_MAC
(CKR_VENDOR_DEFINED + 0x15)	CKR_DAK_MISSING
(CKR_VENDOR_DEFINED + 0x16)	CKR_BAD_DAK
(CKR_VENDOR_DEFINED + 0x17)	CKR_SIM_AUTHORIZATION_FAILED
(CKR_VENDOR_DEFINED + 0x18)	CKR_SIM_VERSION_UNSUPPORTED
(CKR_VENDOR_DEFINED + 0x19)	CKR_SIM_CORRUPT_DATA
(CKR_VENDOR_DEFINED + 0x1a)	CKR_USER_NOT_AUTHORIZED
(CKR_VENDOR_DEFINED + 0x1b)	CKR_MAX_OBJECT_COUNT_EXCEEDED
(CKR_VENDOR_DEFINED + 0x1c)	CKR_SO_LOGIN_FAILURE_THRESHOLD

Code	Name
(CKR_VENDOR_DEFINED + 0x1d)	CKR_SIM_AUTHFORM_INVALID
(CKR_VENDOR_DEFINED + 0x1e)	CKR_CITS_DAK_MISSING
(CKR_VENDOR_DEFINED + 0x1f)	CKR_UNABLE_TO_CONNECT
(CKR_VENDOR_DEFINED + 0x20)	CKR_PARTITION_DISABLED
(CKR_VENDOR_DEFINED + 0x21)	CKR_CALLBACK_ERROR
(CKR_VENDOR_DEFINED + 0x22)	CKR_SECURITY_PARAMETER_MISSING
(CKR_VENDOR_DEFINED + 0x23)	CKR_SP_TIMEOUT
(CKR_VENDOR_DEFINED + 0x24)	CKR_TIMEOUT
(CKR_VENDOR_DEFINED + 0x25)	CKR_ECC_UNKNOWN_CURVE
(CKR_VENDOR_DEFINED + 0x26)	CKR_MTK_ZEROIZED
(CKR_VENDOR_DEFINED + 0x27)	CKR_MTK_STATE_INVALID
(CKR_VENDOR_DEFINED + 0x28)	CKR_INVALID_ENTRY_TYPE
(CKR_VENDOR_DEFINED + 0x29)	CKR_MTK_SPLIT_INVALID
(CKR_VENDOR_DEFINED + 0x2a)	CKR_HSM_STORAGE_FULL
(CKR_VENDOR_DEFINED + 0x2b)	CKR_DEVICE_TIMEOUT
(CKR_VENDOR_DEFINED + 0x2C)	CKR_CONTAINER_OBJECT_STORAGE_FULL
(CKR_VENDOR_DEFINED + 0x2D)	CKR_PED_CLIENT_NOT_RUNNING
(CKR_VENDOR_DEFINED + 0x2E)	CKR_PED_UNPLUGGED
(CKR_VENDOR_DEFINED + 0x2F)	CKR_ECC_POINT_INVALID
(CKR_VENDOR_DEFINED + 0x30)	CKR_OPERATION_NOT_ALLOWED
(CKR_VENDOR_DEFINED + 0x31)	CKR_LICENSE_CAPACITY_EXCEEDED
(CKR_VENDOR_DEFINED + 0x32)	CKR_LOG_FILE_NOT_OPEN
(CKR_VENDOR_DEFINED + 0x33)	CKR_LOG_FILE_WRITE_ERROR

Code	Name
(CKR_VENDOR_DEFINED + 0x34)	CKR_LOG_BAD_FILE_NAME
(CKR_VENDOR_DEFINED + 0x35)	CKR_LOG_FULL
(CKR_VENDOR_DEFINED + 0x36)	CKR_LOG_NO_KCV
(CKR_VENDOR_DEFINED + 0x37)	CKR_LOG_BAD_RECORD_HMAC
(CKR_VENDOR_DEFINED + 0x38)	CKR_LOG_BAD_TIME
(CKR_VENDOR_DEFINED + 0x39)	CKR_LOG_AUDIT_NOT_INITIALIZED
(CKR_VENDOR_DEFINED + 0x3A)	CKR_LOG_RESYNC_NEEDED
(CKR_VENDOR_DEFINED + 0x3B)	CKR_AUDIT_LOGIN_TIMEOUT_IN_PROGRESS
(CKR_VENDOR_DEFINED + 0x3C)	CKR_AUDIT_LOGIN_FAILURE_THRESHOLD
(CKR_VENDOR_DEFINED + 0x3D)	CKR_INVALID_FUF_TARGET
(CKR_VENDOR_DEFINED + 0x3E)	CKR_INVALID_FUF_HEADER
(CKR_VENDOR_DEFINED + 0x3F)	CKR_INVALID_FUF_VERSION
(CKR_VENDOR_DEFINED + 0x40)	CKR_ECC_ECC_RESULT_AT_INF
(CKR_VENDOR_DEFINED + 0x41)	CKR_AGAIN
(CKR_VENDOR_DEFINED + 0x42)	CKR_TOKEN_COPIED
(CKR_VENDOR_DEFINED + 0x43)	CKR_SLOT_NOT_EMPTY
(CKR_VENDOR_DEFINED + 0x44)	CKR_USER_ALREADY_ACTIVATED
(CKR_VENDOR_DEFINED + 0x45)	CKR_STC_NO_CONTEXT
(CKR_VENDOR_DEFINED + 0x46)	CKR_STC_CLIENT_IDENTITY_NOT_CONFIGURED
(CKR_VENDOR_DEFINED + 0x47)	CKR_STC_PARTITION_IDENTITY_NOT_CONFIGURED
(CKR_VENDOR_DEFINED + 0x48)	CKR_STC_DH_KEYGEN_ERROR
(CKR_VENDOR_DEFINED + 0x49)	CKR_STC_CIPHER_SUITE_REJECTED
(CKR_VENDOR_DEFINED + 0x4a)	CKR_STC_DH_KEY_NOT_FROM_SAME_GROUP

Code	Name
(CKR_VENDOR_DEFINED + 0x4b)	CKR_STC_COMPUTE_DH_KEY_ERROR
(CKR_VENDOR_DEFINED + 0x4c)	CKR_STC_FIRST_PHASE_KDF_ERROR
(CKR_VENDOR_DEFINED + 0x4d)	CKR_STC_SECOND_PHASE_KDF_ERROR
(CKR_VENDOR_DEFINED + 0x4e)	CKR_STC_KEY_CONFIRMATION_FAILED
(CKR_VENDOR_DEFINED + 0x4f)	CKR_STC_NO_SESSION_KEY
(CKR_VENDOR_DEFINED + 0x50)	CKR_STC_RESPONSE_BAD_MAC
(CKR_VENDOR_DEFINED + 0x51)	CKR_STC_NOT_ENABLED
(CKR_VENDOR_DEFINED + 0x52)	CKR_STC_CLIENT_HANDLE_INVALID
(CKR_VENDOR_DEFINED + 0x53)	CKR_STC_SESSION_INVALID
(CKR_VENDOR_DEFINED + 0x54)	CKR_STC_CONTAINER_INVALID
(CKR_VENDOR_DEFINED + 0x55)	CKR_STC_SEQUENCE_NUM_INVALID
(CKR_VENDOR_DEFINED + 0x56)	CKR_STC_NO_CHANNEL
(CKR_VENDOR_DEFINED + 0x57)	CKR_STC_RESPONSE_DECRYPT_ERROR
(CKR_VENDOR_DEFINED + 0x58)	CKR_STC_RESPONSE_REPLAYED
(CKR_VENDOR_DEFINED + 0x59)	CKR_STC_REKEY_CHANNEL_MISMATCH
(CKR_VENDOR_DEFINED + 0x5a)	CKR_STC_RSA_ENCRYPT_ERROR
(CKR_VENDOR_DEFINED + 0x5b)	CKR_STC_RSA_SIGN_ERROR
(CKR_VENDOR_DEFINED + 0x5c)	CKR_STC_RSA_DECRYPT_ERROR
(CKR_VENDOR_DEFINED + 0x5d)	CKR_STC_RESPONSE_UNEXPECTED_KEY
(CKR_VENDOR_DEFINED + 0x5e)	CKR_STC_UNEXPECTED_NONCE_PAYLOAD_SIZE
(CKR_VENDOR_DEFINED + 0x5f)	CKR_STC_UNEXPECTED_DH_DATA_SIZE
(CKR_VENDOR_DEFINED + 0x60)	CKR_STC_OPEN_CIPHER_MISMATCH
(CKR_VENDOR_DEFINED + 0x61)	CKR_STC_OPEN_DHNIST_PUBKEY_ERROR

Code	Name
(CKR_VENDOR_DEFINED + 0x62)	CKR_STC_OPEN_KEY_MATERIAL_GEN_FAIL
(CKR_VENDOR_DEFINED + 0x63)	CKR_STC_OPEN_RESP_GEN_FAIL
(CKR_VENDOR_DEFINED + 0x64)	CKR_STC_ACTIVATE_MACTAG_U_VERIFY_FAIL
(CKR_VENDOR_DEFINED + 0x65)	CKR_STC_ACTIVATE_MACTAG_V_GEN_FAIL
(CKR_VENDOR_DEFINED + 0x66)	CKR_STC_ACTIVATE_RESP_GEN_FAIL
(CKR_VENDOR_DEFINED + 0x67)	CKR_CHALLENGE_INCORRECT
(CKR_VENDOR_DEFINED + 0x68)	CKR_ACCESS_ID_INVALID
(CKR_VENDOR_DEFINED + 0x69)	CKR_ACCESS_ID_ALREADY_EXISTS
(CKR_VENDOR_DEFINED + 0x6a)	CKR_KEY_NOT_KEKABLE
(CKR_VENDOR_DEFINED + 0x6b)	CKR_MECHANISM_INVALID_FOR_FP
(CKR_VENDOR_DEFINED + 0x6c)	CKR_OPERATION_INVALID_FOR_FP
(CKR_VENDOR_DEFINED + 0x6d)	CKR_SESSION_HANDLE_INVALID_FOR_FP
(CKR_VENDOR_DEFINED + 0x6e)	CKR_CMD_NOT_ALLOWED_HSM_IN_TRANSPORT
(CKR_VENDOR_DEFINED + 0X6f)	CKR_OBJECT_ALREADY_EXISTS
(CKR_VENDOR_DEFINED + 0X70)	CKR_PARTITION_ROLE_DESC_VERSION_INVALID
(CKR_VENDOR_DEFINED + 0X71)	CKR_PARTITION_ROLE_POLICY_VERSION_INVALID
(CKR_VENDOR_DEFINED + 0X72)	CKR_PARTITION_ROLE_POLICY_SET_VERSION_INVALID
(CKR_VENDOR_DEFINED + 0X73)	CKR_REKEK_KEY
(CKR_VENDOR_DEFINED + 0X74)	CKR_KEK_RETRY_FAILURE
(CKR_VENDOR_DEFINED + 0X75)	CKR_RNG_RESEED_TOO_EARLY
(CKR_VENDOR_DEFINED + 0X76)	CKR_HSM_TAMPERED
(CKR_VENDOR_DEFINED + 0x77)	CKR_CONFIG_CHANGE_ILLEGAL
(CKR_VENDOR_DEFINED + 0x78)	CKR_SESSION_CONTEXT_NOT_ALLOCATED

Code	Name
(CKR_VENDOR_DEFINED + 0x79)	CKR_SESSION_CONTEXT_ALREADY_ALLOCATED
(CKR_VENDOR_DEFINED + 0x7a)	CKR_INVALID_BL_ITB_AUTH_HEADER
(CKR_VENDOR_DEFINED + 0x114)	CKR_OBJECT_READ_ONLY
(CKR_VENDOR_DEFINED + 0x136)	CKR_KEY_NOT_ACTIVE

CHAPTER 19: Updates and Upgrades

Thales Group releases periodic updates to the SafeNet Luna PCIe HSM firmware, as well as updated versions of the SafeNet Luna HSM Client software. If you have recently purchased a new SafeNet Luna PCIe HSM and your organization requires FIPS certification, you can download and install a FIPS-validated version of the HSM firmware. You can download these updates as they become available from the Thales Group Customer Support Portal: https://supportportal.gemalto.com.

Depending on the model of SafeNet Luna PCIe HSM you selected at time of purchase, you may also be able to purchase upgrades to the HSM's capabilities.

The following chapter provides tested update paths and procedures for installing update packages, as well as a list of the version dependencies for certain features. It contains the following sections:

> "Update Considerations" below
> "Version Dependencies by Feature" on page 317
> "Updating the SafeNet Luna HSM Client" on page 318
> "Updating the SafeNet Luna PCIe HSM or SafeNet Luna Backup HSM Firmware" on page 318
> "Rolling Back the SafeNet Luna HSM Firmware" on page 319
> "Upgrading HSM Capabilities" on page 320

Update Considerations

Before you install any of the updates, consider the following guidelines:

> Back up all important cryptographic material.
> Stop all client applications running cryptographic operations on the HSM.
> Use an uninterruptible power supply (UPS) to power your HSM. There is a small chance that a power failure during an update could leave your HSM in an unrecoverable condition.

Valid Update Paths

The following table provides tested paths for updating to the current software/firmware versions.

Component	Directly from version	To version
SafeNet Luna HSM Client software	Any	7.3

Component	Directly from version	To version
SafeNet Luna HSM firmware	7.0.1, 7.0.2	7.0.3, 7.2.0
	7.1.0	7.2.0
	7.0.3, 7.2.0	7.3.0
SafeNet Backup HSM firmware	6.10.9, 6.26.0	6.27.0
SafeNet Luna PED firmware	2.7.1	N/A
	2.8.0	N/A

FIPS-Validated Firmware Versions

The following firmware versions are all FIPS-140-2 Level 3 certified per certificate #3205:

https://csrc.nist.gov/projects/cryptographic-module-validation-program/Certificate/3205

- Luna firmware v. 7.0.3 (recommended)
- Luna firmware v. 7.0.2 (see F5 note, below)
- Luna firmware v. 7.0.1

Recommended Minimum Versions

Generally, Thales Group recommends that you always keep your HSM firmware and client software up to date, to benefit from the latest features and bug fixes. If regular updates are not possible or convenient, the following table lists the recommended minimum firmware and software versions for use with SafeNet Luna 7 HSMs. If you are running an earlier version, Thales Group advises upgrading to the version(s) below (or later) to ensure that you have critical bug fixes and security updates.

	Luna HSM Client	Luna HSM Firmware
SafeNet Luna PCIe HSM 7 Minimum Recommended Configuration	7.2	7.2.0
		7.0.3

NOTE Customers who wish to use Luna 7 HSMs with F5 Network BIG-IP 13.1 appliances should follow F5 guidelines for Supported SafeNet client and HSM versions (https://support.f5.com/kb/en-us/products/big-ip_ltm/manuals/product/f5-safenet-hsm-version-interoperability-matrix.html). At the time of this release, F5's supported versions for Luna 7 are Luna HSM Client 7.1 with firmware 7.0.2.

Version Dependencies by Feature

Some of the SafeNet Luna PCIe HSM functionality described in the documentation has been introduced in updates since the initial product release. For your own reasons, you may wish to apply some aspects of a product update and not others. For example:

- you may choose to update client software while keeping an earlier, FIPS-certified firmware version
- if you are maintaining a large number of client workstations, it may be cumbersome to apply software updates to all of them

The following table outlines the SafeNet Luna PCIe HSM functions that depend on a certain software/firmware version, or have other requirements you must consider.

Function	Minimum Version Requirements	Notes
Improved Luna HSM Client > Version-Compatible Luna HSM Client (Luna HSMs version 6.2.1 and higher) > "Modifying the installed Luna HSM client software" on page 1 > User-Defined Luna HSM Client install paths	**Client:** 7.2	> The **PE1756Enabled** setting on Luna 6.x HSMs is not supported for use with the Version-Compatible Luna HSM Client > Minimum OS requirements for Luna HSM Client 7.2 must be met (Refer to the CRN for details)
Relabel partitions > "partition changelabel" on page 1	**Firmware:** 7.2.0 **Client:** 7.2	
Crypto User can clone public objects	**Firmware:** 7.2.0	The Crypto User (CU) role has always been able to create public objects, but not clone them. In HA mode, this would cause the replication and subsequent object creation operations to fail. Firmware 7.2.0 allows the CU to clone public objects, and therefore to perform operations on HA groups without Crypto Officer authentication.
Configure partition policies for export of private keys > "Keys In Hardware vs. Private Key Export" on page 172	**Firmware:** 7.1.0	You can configure partition policies for Cloning or Key Export Mode manually, as long as you have updated the HSM firmware. To set these modes using Policy Templates, you must meet the Policy Template requirements.
Policy Templates > "Policy Templates" on page 93	**Firmware:** 7.1.0 **Client:** 7.1	

Updating the SafeNet Luna HSM Client

To update the SafeNet Luna HSM Client software, first uninstall any previous version of the Client. Then, run the new installer the same way you performed the original installation (refer to "SafeNet Luna HSM Client Software Installation" on page 1).

On Windows systems, the client uninstaller removes libraries, utilities, and other material related to the client, but does not remove configuration files and certificates. This allows you to install the newer version and resume operations without having to manually restore configuration settings and re-register client and appliance NTLS certificates.

Updating the SafeNet Luna PCIe HSM or SafeNet Luna Backup HSM Firmware

To update the firmware on a SafeNet Luna PCIe HSM or SafeNet Luna Backup HSM, download the desired firmware version from the Thales Group Support Portal. Use LunaCM on the host workstation to apply the update. You require:

> SafeNet Luna HSM firmware update file (<filename>**.fuf**) and/or

> SafeNet Luna Backup HSM firmware update file (<filename>**.fuf**)

> the firmware update authentication code file(s) (<filename>**.txt**)

> **CAUTION!** Use an uninterruptible power supply (UPS) to power your HSM. There is a small chance that a power failure during an update could leave your HSM in an unrecoverable condition.

To update the SafeNet Luna PCIe HSM or SafeNet Luna Backup HSM firmware

1. Copy the firmware file (<filename>**.fuf**) and the authentication code file (<filename>**.txt**) to the SafeNet Luna HSM Client root directory.
   - Windows: C:\Program Files\SafeNet\LunaClient
   - Linux: /usr/safenet/lunaclient/bin
   - Solaris: /opt/safenet/lunaclient/bin

   > **NOTE** On some Windows configurations, you might not have authority to copy or unzip files directly into **C:\Program Files\...**. If this is the case, put the files in a known location that you can reference in a LunaCM command.

2. Launch LunaCM.
3. If more than one HSM is installed, set the active slot to the Admin partition of the HSM you wish to update ("slot set" on page 1).

   lunacm:>**slot set -slot** <slot_number>
4. Log in as HSM SO ("role login" on page 1).

   lunacm:>**role login -name so**

5. Apply the new firmware update by specifying the update file and the file containing the authorization code. If the files are not located in the SafeNet Luna Network HSM Client directory, specify the filepaths ("hsm updatefw" on page 1).

   lunacm:>**hsm updatefw -fuf** <filename>**.fuf -authcode** <filename>**.txt**

Changing the Firmware Upgrade Permissions (Linux only)

By default, the root user and any user who is part of the **hsmusers** group can perform a firmware update. You can use this procedure to restrict firmware update operations to root only (that is, disable firmware update for members of the **hsmusers** group).

To restrict firmware update operations to the root user only

1. Open the the **/etc/modprobe.d/k7.conf** file for editing:

   sudoedit /etc/modprobe.d/k7.conf

2. Change the **k7_rootonly_reset** option from **0** to **1**. Save the file and exit the editor.
3. Stop any processes that are using the K7 driver. Typically this means stopping the **pedclient** service, and the **luna-snmp** service, if you are using SNMP.

   sudo systemctl stop pedclient_service

   sudo systemctl stop luna-snmp

4. Reload the driver:

   sudo systemctl reload k7

Rolling Back the SafeNet Luna HSM Firmware

When updating the HSM firmware, the SafeNet Luna PCIe HSM saves the previously-installed firmware version on the HSM. If required, you can roll back to this previously-installed version. Rollback allows you to try firmware without permanently committing to the new version.

Rollback does not create a new rollback target; a single rollback target is preserved when a firmware update is performed. After a rollback operation, no further rollback is possible until the next firmware update saves the pre-update version as the new rollback target.

> **CAUTION!** Firmware rollback is destructive; earlier firmware versions might have older mechanisms and security vulnerabilities that a new version does not. Back up any important materials before rolling back the firmware. This procedure zeroizes the HSM and all cryptographic objects are erased.

To roll back the SafeNet Luna HSM firmware to the previous version

1. Check the previous firmware version that is available on the HSM ("hsm showinfo" on page 1).

   lunacm:>**hsm showinfo**

2. Back up any important cryptographic objects currently stored on the HSM (see "Backup and Restore" on page 34).

3. At the LunaCM prompt, login as HSM SO ("role login" on page 1).

   lunacm:>**role login -name so**

4. Roll back the HSM firmware ("hsm rollbackfw" on page 1).

   lunacm:>**hsm rollbackfw**

   LunaCM performs an automatic restart following the rollback procedure.

5. Re-initialize the HSM and restore your partition from backup.

Upgrading HSM Capabilities

A Secure Capability Upgrade for SafeNet Luna PCIe HSM is delivered to you as a downloaded file set. Follow the FTP instructions in the email you received from Thales Group Technical Support and unzip the files to the host workstation. The update procedure is similar to the procedure for firmware updates.

> **NOTE** On some Windows configurations, you might not have authority to copy or unzip files directly into **C:\Program Files\...**. If this is the case, put the files in a known location that you can reference in a LunaCM command.

You require:

> the SafeNet Luna PCIe HSM capability upgrade file (<filename>**.cuf**)

> the capability update authentication code file (<filename>**.txt**)

Installing the Capability Upgrade

Once the files are unpacked and available on the host workstation, open a command-prompt session.

To install the upgrade package

1. Navigate to the SafeNet Luna HSM Client directory and launch LunaCM.
2. Log in as HSM SO ("role login" on page 1).

   lunacm:>**role login -name so**

3. Apply the new capability by specifying the upgrade file and the file containing the authorization code. If the files are not located in the SafeNet Luna Network HSM Client directory, specify the filepaths ("hsm updatecap" on page 1).

   lunacm:>**hsm updatecap -cuf** <upgrade_file> **-authcode** <authcode_file>

4. Check that the new capability is in place ("hsm showpolicies" on page 1).

   lunacm:>**hsm showpolicies**

CHAPTER 20: Users and Roles

The security of an HSM and its cryptographic contents depends on well-controlled access to that HSM. A controlled access policy is defined by:

- the set of users with valid login credentials for the host system, the HSM and the application partition
- the actions each user is allowed to perform when logged in (the user's role)

For example, an access policy that adheres to the PKCS#11 standard requires two roles: the security officer (SO), who administers the user account(s), and the standard user, who performs cryptographic operations. When a user logs in to the HSM, they can perform only those functions that are permitted for their role.

Access to SafeNet Luna PCIe HSM is controlled through an enhanced version of the PKCS#11 hierarchy of roles, assigned to different users in your organization. Each role allows its user to execute a different set of commands to perform specialized tasks at one of the following levels:

- "HSM Roles and Procedures" on the next page
- "Partition Roles and Procedures" on page 325

HSM-Level Roles

HSM roles are responsible for administration, configuration, and auditing of the HSM. These roles log in to the SafeNet Luna PCIe HSM Admin partition using LunaCM. HSM-level roles cannot perform cryptographic operations on the application partition. See "HSM Roles and Procedures" on the next page for details.

Table 1: HSM Roles

HSM Security Officer (SO) **PED Key: Blue**	> Initializes the HSM, creating the SO credential > Creates/deletes the application partition > Configures global HSM policies > Performs updates of the HSM firmware
Auditor (AU) **PED Key: White**	> Manages HSM audit logging

Partition-Level Roles

Partition-level roles are responsible for administration and configuration of the application partition, and using the partition to perform cryptographic functions. Partition roles log in using LunaCM, or supply their credentials via crypto applications. See "Partition Roles and Procedures" on page 325 for details.

Table 2: Partition Roles

Partition Security Officer (PO) **PED Key: Blue**	> Initializes the partition, creating the PO credential and setting the cloning domain > Initializes the Crypto Officer role and can reset the CO credential (if permitted by HSM policy) > Configures partition policies
Crypto Officer (CO) **PED Key: Black**	> Creates and modifies cryptographic objects on the partition > Manages backup and restore operations for the partition > Performs cryptographic functions via user applications > Initializes the Crypto User role and can reset the CU credential
Crypto User (CU) **PED Key: Gray**	> Performs cryptographic functions via user applications (optional read-only role) > Can create public objects only > Can perform backup/restore of public objects on the partition

HSM Roles and Procedures

SafeNet Luna PCIe HSM divides roles on the HSM according to an enhanced version of the PKCS#11 standard. Configuration, administration, and auditing of the HSM itself is the responsibility of the roles described below. Cryptographic functions take place on the application partition, which has a different set of independent roles (see "Partition Roles and Procedures" on page 325).

Personnel holding the HSM roles described below access HSM functions by logging in to the Admin partition on the HSM using LunaCM. They must therefore have the appropriate Administrator access to the workstation hosting the SafeNet Luna PCIe HSM.

The HSM-level roles are as follows:

HSM Security Officer (SO)

The HSM SO handles all administrative and configuration tasks on the HSM, including:

> Initializing the HSM and setting the SO credential (see "HSM Initialization" on page 160)

> Setting and changing global HSM policies (see "HSM Capabilities and Policies" on page 82)

> Creating/deleting the application partition (see "Create Application Partitions" on page 1)

> Updating the HSM firmware (see "Updating the SafeNet Luna PCIe HSM or SafeNet Luna Backup HSM Firmware" on page 318)

Managing the HSM Security Officer Role

Refer also to the following procedures to manage the HSM SO role:

> "Logging In as HSM Security Officer" on the next page

> "Changing a Role Credential" on page 328

> "Failed HSM Logins" on page 329

Auditor (AU)

The Auditor is responsible for managing HSM audit logging. These responsibilities have been separated from the other roles on the HSM and application partition so that the Auditor can provide independent oversight of all HSM processes, and no other user, including the HSM SO, can clear those logs. The Auditor's tasks include:

- Initializing the Auditor role
- Setting up audit logging on the HSM
- Configuring the maximum size of audit log files and the time interval for log rotation
- Archiving the audit logs

Managing the Auditor Role

Refer to "Configuring and Using Audit Logging" on page 22 for procedures involving the Auditor role. See also:

- "Logging In as Auditor" below
- "Changing a Role Credential" on page 328
- "Failed HSM Logins" on page 329

Administrator (AD)

The HSM Administrator is a deprecated role on the Admin partition whose functions are now served by the application partition roles (see "Partition Roles and Procedures" on page 325). Initializing this role is not recommended.

Logging In as HSM Security Officer

Before you can create an application partition or perform other administrative functions on the HSM, you must log in to the SafeNet Luna PCIe HSM's Admin partition as HSM Security Officer (SO), or administrative commands will fail.

To log in as HSM SO

1. Launch LunaCM on the SafeNet Luna PCIe HSM host workstation.
2. Set the active slot to the HSM Admin partition ("slot set" on page 1).

   lunacm:>**slot set -slot** <slotnum>
3. Log in as HSM SO (see "role login" on page 1).

   lunacm:>**role login -name so**

   You are prompted for the HSM SO credential.

Logging In as Auditor

Before you can change the audit logging configuration, archive audit logs, or verify audit logs from another HSM, you must log in to the SafeNet Luna PCIe HSM's Admin partition as Auditor (AU), or relevant commands will fail.

To log in as Auditor

1. Launch LunaCM on the SafeNet Luna PCIe HSM host workstation.
2. Set the active slot to the HSM Admin partition ("slot set" on page 1).

   lunacm:>**slot set -slot** <slotnum>
3. Log in as Auditor (see "role login" on page 1).

   lunacm:>**role login -name au**

   You are prompted for the Auditor credential.

Changing a Role Credential

From time to time, you may need to change the credential for a role. The credential might have been compromised, or your organization's security policy may mandate password changes after a specific time interval. The following procedure allows you to change the credential for a role (HSM SO, Auditor, Partition SO, Crypto Officer, Crypto User). You must first log in using the role's current credential.

To change a role credential

1. In LunaCM, log in using the role's current credential (see "Logging In to the Application Partition" on page 327).
2. Change the credential for the logged-in role ("role changepw" on page 1). If you are using a password-authenticated partition, specify a new password. If you are using a PED-authenticated partition, ensure that you have a blank or rewritable PED key available. Refer to "Creating PED Keys" on page 212 for details on creating PED keys.

   lunacm:>**role changepw -name** <role>

```
lunacm:> role changepw -name co

        enter existing password: ********

        enter new password: ********

        re-enter new password: ********

Command Result : No Error
```

3. To change the CO or CU challenge secret for an Activated PED-authenticated partition, specify the **-oldpw** and/or **-newpw** options ("role changepw" on page 1).

   lunacm:>**role changepw -name** <role> **-oldpw** <oldpassword> **-newpw** <newpassword>

```
lunacm:> role changepw -name co -oldpw PASSWORD -newpw userpin

        This role has secondary credentials.
        You are about to change the secondary credentials.
        Are you sure you wish to continue?


        Type 'proceed' to continue, or 'quit' to quit now -> proceed
```

```
Command Result : No Error
```

Partition Roles and Procedures

All cryptographic operations take place on an application partition. This partition is created on the SafeNet Luna PCIe HSM by the HSM SO and is designed to function independently of the Admin partition, with its own Security Officer and users. This provides more flexibility in meeting the security needs of your organization. Personnel holding the roles described below must have administrative access to the SafeNet Luna PCIe HSM host workstation.

The partition-level roles are as follows:

Partition Security Officer (PO)

The Partition SO handles all administrative and configuration tasks on the application partition, including:

- Initializing the partition, setting the PO credential, and setting a cloning domain for the partition (see "Creating an Application Partition on the HSM" on page 1)
- Configuring partition policies (see "Partition Capabilities and Policies" on page 87)
- Initializing the Crypto Officer role (see "Initializing the Crypto Officer Role" on the next page)
- Activating the partition (see "Activation and Auto-Activation on PED-Authenticated Partitions" on page 178)

Managing the Partition SO Role

Refer also to the following procedures to manage the PO role:

- "Logging In to the Application Partition" on page 327
- "Changing a Role Credential" on page 328
- "Failed Partition Logins" on page 330

Crypto Officer (CO)

The Crypto Officer is the primary user of the application partition and the cryptographic objects stored on it. The Crypto Officer has the following responsibilities:

- Creating, deleting, and modifying cryptographic objects via user applications
- Performing cryptographic operations via user applications
- Managing backup and restore operations for partition objects (see "Backup and Restore" on page 34)
- Initializing the Crypto User role (see "Initializing the Crypto User Role" on page 327)

Managing the Crypto Officer Role

Refer also to the following procedures to manage the CO role:

- "Logging In to the Application Partition" on page 327
- "Changing a Role Credential" on page 328
- "Failed Partition Logins" on page 330

Crypto User (CU)

The Crypto User is an optional role that can perform cryptographic operations using partition objects in a read-only capacity, but can create only public objects. This role is useful in that it provides limited access; the Crypto Officer is the only role that can make significant changes to the contents of the partition. The Crypto User has the following capabilities:

- Performing operations like encrypt/decrypt and sign/verify using objects on the partition
- Creating and backing up public objects (see "Backup and Restore" on page 34)

Managing the Crypto User Role

Refer also to the following procedures to manage the CU role:

- "Logging In to the Application Partition" on the next page
- "Changing a Role Credential" on page 328
- "Failed Partition Logins" on page 330

Initializing the Crypto Officer and Crypto User Roles

The following procedures will allow you to initialize the Crypto Officer (CO) and Crypto User (CU) roles and set an initial credential.

Initializing the Crypto Officer Role

The Crypto Officer (CO) is the primary user of the application partition and the cryptographic objects stored on it. The Partition Security Officer (PO) must initialize the CO role and assign an initial credential.

To initialize the Crypto Officer role

1. In LunaCM, log in to the partition as Partition SO (see "Logging In to the Application Partition" on the next page).

   lunacm:>**role login -name po**

2. Initialize the Crypto Officer role ("role init" on page 1). If you are using a password-authenticated partition, specify a CO password. If you are using a PED-authenticated partition, ensure that you have a blank or rewritable black PED key available. Refer to "Creating PED Keys" on page 212 for details on creating PED keys.

   lunacm:>**role init -name co**

   ```
   lunacm:> role init -name co

           enter new password: ********

           re-enter new password: ********

   Command Result : No Error
   ```

3. Provide the CO credential to your designated Crypto Officer.

> **NOTE** If **HSM policy 21: Force user PIN change after set/reset** is enabled, the CO must change the credential before any other actions are permitted. See "Changing a Role Credential" on the next page.

Initializing the Crypto User Role

The Crypto User (CU) is an optional role that can perform cryptographic operations using partition objects in a read-only capacity, but can only create public objects. The Crypto Officer must initialize the CO role and assign an initial credential.

To initialize the Crypto User role

1. In LunaCM, log in to the partition as Crypto Officer (see "Logging In to the Application Partition" below).

   lunacm:>**role login -name co**

2. Initialize the Crypto User role ("role init" on page 1). If you are using a password-authenticated partition, specify a CU password. If you are using a PED-authenticated partition, ensure that you have a blank or rewritable gray PED key available. Follow the instructions on the Luna PED screen. Refer to "Creating PED Keys" on page 212 for details on creating PED keys.

   lunacm:>**role init -name cu**

   ```
   lunacm:> role init -name cu

           enter new password: ********

           re-enter new password: ********

   Command Result : No Error
   ```

3. Provide the CU credential to your designated Crypto User.

> **NOTE** If **HSM policy 21: Force user PIN change after set/reset** is enabled, the CU must change the credential before any other actions are permitted. See "Changing a Role Credential" on the next page.

Logging In to the Application Partition

Before you can perform administrative tasks on the partition or its stored cryptographic objects, you must log in with the appropriate role:

- Partition Security Officer (specify **po** for <role>)
- Crypto Officer (specify **co** for <role>)
- Crypto User (specify **cu** for <role>)

To log in to the application partition

1. Launch LunaCM on the SafeNet Luna PCIe HSM host workstation.
2. Set the active slot to the desired partition ("slot set" on page 1).

lunacm:>**slot set -slot** <slotnum>

3. Log in by specifying your role on the partition ("role login" on page 1).

   lunacm:>**role login -name** <role>

   You are prompted for the role's credential.

Changing a Role Credential

From time to time, you may need to change the credential for a role. The credential might have been compromised, or your organization's security policy may mandate password changes after a specific time interval. The following procedure allows you to change the credential for a role (HSM SO, Auditor, Partition SO, Crypto Officer, Crypto User). You must first log in using the role's current credential.

To change a role credential

1. In LunaCM, log in using the role's current credential (see "Logging In to the Application Partition" on the previous page).
2. Change the credential for the logged-in role ("role changepw" on page 1). If you are using a password-authenticated partition, specify a new password. If you are using a PED-authenticated partition, ensure that you have a blank or rewritable PED key available. Refer to "Creating PED Keys" on page 212 for details on creating PED keys.

   lunacm:>**role changepw -name** <role>

```
lunacm:> role changepw -name co

        enter existing password: ********

        enter new password: ********

        re-enter new password: ********

Command Result : No Error
```

3. To change the CO or CU challenge secret for an Activated PED-authenticated partition, specify the **-oldpw** and/or **-newpw** options ("role changepw" on page 1).

   lunacm:>**role changepw -name** <role> **-oldpw** <oldpassword> **-newpw** <newpassword>

```
lunacm:> role changepw -name co -oldpw PASSWORD -newpw userpin

        This role has secondary credentials.
        You are about to change the secondary credentials.
        Are you sure you wish to continue?


        Type 'proceed' to continue, or 'quit' to quit now -> proceed


Command Result : No Error
```

Resetting the Crypto Officer or Crypto User Credential

If necessary, the Crypto Officer can reset the Crypto User credential at any time, without providing the current credential. This is useful in cases where the Crypto User credential has been lost or otherwise compromised.

Prerequisites for Crypto Officer Reset

The Partition SO can also reset the Crypto Officer's credential, if **HSM policy 15: Enable SO reset of partition PIN** is enabled. By default, this policy is not enabled, and changing it is destructive. If you want the Partition SO to be able to reset the CO's credential, the HSM SO must enable this policy before creating the application partition (see "Partition Capabilities and Policies" on page 87).

> **CAUTION!** HSM policy 15 is destructive when turned on. All partitions on the HSM and their contents will be erased.

To reset the Crypto Officer or Crypto User credential

1. Log in with the appropriate role (see "Logging In to the Application Partition" on page 327).
2. Reset the desired role's credential ("role resetpw" on page 1).

   lunacm:>**role resetpw -name** <role>

   You are prompted to set a new credential for the role.
3. Provide the new credential to the Crypto Officer or Crypto User.

> **NOTE** If **HSM policy 21: Force user PIN change after set/reset** is enabled, the user must change the credential before any other actions are permitted. See "Changing a Role Credential" on the previous page.

Failed Login Attempts

The consequences of multiple failed login attempts vary by role, depending on the severity of the security risk posed by that role being compromised. This is a security feature meant to thwart repeated, unauthorized attempts to access your cryptographic material.

> **NOTE** The system must actually receive some erroneous/false information before it logs a failed attempt, like an incorrect password -- if you merely forget to insert a PED key, or inserted the wrong color key, that is not counted as a failed attempt. You must insert an incorrect PED key of the correct type, or enter an incorrect PED PIN or challenge secret, to fail a login attempt.

> "Failed HSM Logins" below
> "Failed Partition Logins" on the next page
> "Failed Domain or RPV Authentication" on the next page

Failed HSM Logins

At the HSM level, multiple failed logins have the following consequences:

HSM Security Officer

If you fail three (3) consecutive HSM SO login attempts, application partitions are destroyed, the HSM is zeroized and all of its contents are rendered unrecoverable. The number is not adjustable. As soon as you authenticate successfully, the counter is reset to zero.

Auditor

If you fail ten (10) consecutive Auditor login attempts, the Auditor role is locked out for ten minutes.

Failed Partition Logins

At the application partition level, multiple failed logins have the following consequences:

Partition Security Officer

If you fail ten consecutive Partition SO login attempts, the partition is zeroized and all cryptographic objects are destroyed. The Partition SO must re-initialize the partition and Crypto Officer role, who can restore key material from a backup device.

Crypto Officer

If you fail ten consecutive Crypto Officer login attempts, the CO and CU roles are locked out. The default lockout threshold of 10 is governed by partition policy 20: Max failed user logins allowed, and the Partition SO can set this threshold lower if desired (see "Partition Capabilities and Policies" on page 87). Recovery depends on the setting of **HSM policy 15: Enable SO reset of partition PIN**:

- If HSM policy 15 is set to **1** (enabled), the CO and CU roles are locked out. The Partition SO must unlock the CO role and reset the credential (see "Resetting the Crypto Officer or Crypto User Credential" on the previous page).
- If HSM policy 15 is set to **0** (disabled), the CO and CU roles are permanently locked out and the partition contents are no longer accessible. The Partition SO must re-initialize the partition and the Crypto Officer role, who can restore key material from a backup. This is the default setting.

> **CAUTION!** If this is not the desired outcome, ensure that the HSM SO enables this destructive policy before creating and assigning partitions to clients.

Crypto User

If you fail ten consecutive Crypto User login attempts, the CU role is locked out. The default lockout threshold of 10 is governed by partition policy 20: Max failed user logins allowed, and the Partition SO can set this threshold lower if desired (see "Partition Capabilities and Policies" on page 87). The Crypto Officer must unlock the CU role and reset the credential (see "Resetting the Crypto Officer or Crypto User Credential" on the previous page).

Failed Domain or RPV Authentication

If you fail an attempt to authenticate a cloning domain or Remote PED Vector, the consequences are less severe:

Domain

The operation fails. Usually this would be an attempt to back up or restore partitions. Reattempt with the correct domain authentication secret.

Remote PED Vector

The Remote PED setup operation fails. Reattempt with the correct RPV authentication secret (orange PED key).