THALES

SafeNet Luna Network HSM 7.1

ADMINISTRATION GUIDE

Document Information

Product Version	7.1
Document Part Number	007-013576-003
Release Date	13 December 2019

Revision History

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

Trademarks, Copyrights, and Third-Party Software

Table 1: Third-party software used in this product

Software	License and copyright
editline	This product incorporates editline licensed under Apache v2.0 Open Software. Copyright 1992,1993 Simmule Turner and Rich Salz. All rights reserved. You can obtain the full text of the Apache v2.0 Open Software license at the following URL: https://www.apache.org/licenses/LICENSE-2.0
libFDT	Dual License Choice of BSD or GPL-2.0 Copyright (C) 2006 David Gibson, IBM Corporation.
libsodium	ISC License (ISCL) Copyright (C) 2013-2016
Linux Kernel	GPL-2.0
OpenSSH	This product uses a derived version of OpenSSH Copyright 1995 Tatu Ylonen , Espoo, Finland. All rights reserved . Copyright 1995, 1996 by David Mazieres . Copyright 1983, 1990, 1992, 1993, 1995 The Regents of the University of California. All rights reserved You can obtain the full text of the OpenSSH license at the following URL: https://www.openbsd.org/policy.html

Software	License and copyright
OpenSSL	SSLeay License Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com) OpenSSL license Copyright (C) 1998-2002 The OpenSSL Project
Software implementation of SHA2	Proprietary license Copyright (C) 2002, Dr Brian Gladman, Worcester, UK.
Software implementation of AES	Proprietary license Copyright (C) 2001, Dr Brian Gladman <brg@gladman.uk.net>, Worcester, UK.

Disclaimer

All information herein is either public information or is the property of and owned solely by Thales and/or its subsidiaries who shall have and keep the sole right to file patent applications or any other kind of intellectual property protection in connection with such information.

Nothing herein shall be construed as implying or granting to you any rights, by license, grant or otherwise, under any intellectual and/or industrial property rights of or concerning any of Thales's information.

This document can be used for informational, non-commercial, internal, and personal use only provided that:

- The copyright notice, the confidentiality and proprietary legend and this full warning notice appear in all copies.
- This document shall not be posted on any publicly accessible network computer or broadcast in any media, and no modification of any part of this document shall be made.

Use for any other purpose is expressly prohibited and may result in severe civil and criminal liabilities.

The information contained in this document is provided "AS IS" without any warranty of any kind. Unless otherwise expressly agreed in writing, Thales makes no warranty as to the value or accuracy of information contained herein.

The document could include technical inaccuracies or typographical errors. Changes are periodically added to the information herein. Furthermore, Thales reserves the right to make any change or improvement in the specifications data, information, and the like described herein, at any time.

Thales hereby disclaims all warranties and conditions with regard to the information contained herein, including all implied warranties of merchantability, fitness for a particular purpose, title and non-infringement. In no event shall Thales be liable, whether in contract, tort or otherwise, for any indirect, special or consequential 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.

All intellectual property is protected by copyright. All trademarks and product names used or referred to are the copyright of their respective owners. No part of this document may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, chemical, photocopy, recording or otherwise without the prior written permission of Thales Group.

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

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 16
- "Backup and Restore HSMs and Partitions" on page 39
- "Capabilities and Policies" on page 82
- "Configuration File Summary" on page 103
- "Decommissioning, Zeroizing, or Resetting an HSM to Factory Conditions" on page 120
- "High-Availability (HA) Configuration and Operation" on page 128
- "HSM Initialization" on page 165
- "HSM Status Values" on page 175
- "Partitions" on page 177
- "PED Authentication" on page 194
- "Performance" on page 268
- "Security Effects of Administrative Actions" on page 270
- "Secure Transport Mode" on page 275
- "Secure Trusted Channel (STC)" on page 279
- "Slot Numbering and Behavior" on page 301
- "Software, Firmware, and Capability Upgrades" on page 304
- "SNMP Monitoring" on page 311
- "Tamper Events" on page 326
- "Troubleshooting" on page 329
- "User and Password Administration" on page 362

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 14

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

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 HSM appliance, 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 Network 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 Network 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 Network 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

The audit logging function is controlled by two roles on SafeNet Luna Network HSM, that must be used together:

> The "audit" appliance account (use SSH or PuTTy to log in as "audit", instead of "admin", or "operator", or "monitor", etc.)

> The "audit" HSM account (accessible only if you have logged into the appliance as "audit"; this account must be initialized)

On SafeNet Luna Network HSM, the audit logging is managed by an audit user (an appliance system role), in combination with the HSM audit role, through a set of LunaSH commands. The audit user can perform only the audit-logging related tasks and self-related tasks. Other HSM appliance users, such as admin, operator, and monitor, have no access to the audit logging commands.

A default appliance (LunaSH) audit user is automatically created, but must be enabled. Upon first login, the audit user is asked to change their password. That appliance audit user would need to initialize the HSM audit role first, before being able to administer the audit logging. The SafeNet Luna Network HSM admin user can create more audit users when necessary.

To simplify configuration,

> The maximum log file size is capped at 4 MB.

> The log path is kept internal.

> The rotation offset is set at 0.

Audit User on the Appliance

The appliance audit user is a standard user account on SafeNet Luna Network HSM, with default password "PASSWORD" (without the quotation marks). By default, the appliance audit user is disabled. Therefore, you must enable it in LunaSH before it becomes available. See "user enable" on page 1 for the command syntax.

Audit Role on the HSM

A SafeNet Luna Network 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 LunaSH command **audit init** to initialize the audit role, as described in "audit init" on page 1.

Password-authenticated HSMs

For SafeNet Luna Network 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 Network 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.

Appliance Audit User Available Commands

The Audit role has a limited set of operations available to it, on the HSM, as reflected in the reduced command set available to the "audit" user when logged in to the shell (LunaSH).

```
login as: audit
audit@192.20.11.78's password:
Last login: Fri Mar 31 09:37:53 2017 from 10.124.0.31

Luna SA 7.0.0 Command Line Shell - Copyright (c) 2001-2017 SafeNet, Inc. All rights reserved.

lunash:>help

The following top-level commands are available:

 Name                    (short)    Description
 -------------------------------------------------------------------------------
 help                    he         Get Help
 exit                    e          Exit Luna Shell
 hsm                     hs         > Hsm
 audit                   a          > Audit
 my                      m          > My
 network                 n          > Network
```

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

Offset	Length (Chars)	Description
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,
2600001003600B00EA552950140030005D5800000300008001000000000000000000000000000000000000
```

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 show**, 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 Network 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" on the next page

> "Copying Log Files Off the Appliance" on page 27

> "Exporting the Audit Logging Secret and Importing to a Verifying HSM" on page 27

> "Deciphering the Audit Log Records" on page 29

> "Audit Role Authentication Considerations" on page 29

Configuring Audit Logging

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

Prerequisites (HSM SO)

1. Configure the SafeNet Luna Network HSM appliance to use the network time protocol (NTP). See "Timestamping – NTP and Clock Drift" on page 1 in the *Appliance Administration Guide*.
2. Log in to LunaSH as an admin-level user, and enable the audit user. The audit user is necessary to access and work with logs through the LunaSH interface. It is restricted from administrative functions:

   lunash:> **user enable -username audit**

To configure audit logging (Auditor)

1. Using an SSH connection (or a local serial connection), login to LunaSH on the SafeNet Luna Network HSM appliance as **audit** (not as **admin**), using the password "PASSWORD".

   The first time you login as **audit**, you are prompted to change the password to something more secure. To fulfill the purpose of the Audit role, keep the **audit** user's password separate from, and unknown to, the HSM Security Officer:

   The audit user sees a reduced subset of commands suitable to the audit role, only, as follows:

```
Name                 (short)    Description
--------------------------------------------------------------------------------
init                 i          Initialize the Audit role
changePwd            ch         Change Audit User Password or PED Key
login                logi       Login as the Audit user
logout               logo       Logout the Audit user
config               co         Set Audit Parameters
sync                 sy         Synchronize HSM Time to Host Time
show                 sh         Display the Audit logging info
log                  l          > Manage Audit Log Files
secret               se         > Export/Import Audit Logging Secret
remotehost           r          > Configure Audit Logging Remote Hosts
```

> **NOTE** The audit user's commands are not available to the admin user. The audit user has no administrative control over the SafeNet Luna Network HSM appliance. This is a first layer in the separation of roles. This separation allows a user with no administrative control of the appliance and HSM to have oversight of the HSM logs, while also ensuring that an administrator cannot clear those logs.

2. Initialize the **audit** role on the HSM. This enables logging for all subsequent actions performed by the SO and partition user(s):

   lunash:> **audit init**

   - On password-authenticated HSMs, you are prompted for the password and cloning domain.
   - On PED-authenticated HSMs, you are referred to Luna PED, which prompts you for the domain (red PED key) and Audit authentication (white PED key).

3. Now that the audit role exists on the HSM, you can configure the auditing function. However, before you can configure audit logging you must log into the HSM as the **audit** role:

   lunash:> **audit login**

   - On password-authenticated HSMs, you are prompted to enter the password for the audit role.
   - On PED-authenticated HSMs, you are referred to Luna PED, which prompts for the white PED key for the audit role.

   > **NOTE** You are now logged into the appliance as the **audit** user and into the HSM (within the appliance) as the **audit** role. Both are required. The **audit** commands, including HSM login as the **audit** role do not appear if you are logged in as any other named appliance-level user.

4. 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:

   lunash:> **audit sync**

5. 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:

   lunash:> **audit config -parameter event -value** <event_value>

   > **NOTE** The first time you configure audit logging, we suggest using only the **?** option, to see all the available options in the configuration process. See also "audit config " on page 1 in the *LunaSH 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 Audit Officer 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 Audit Officers 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 -parameter event -value all** will log everything the HSM does. This might be useful in some circumstances, but will quickly fill up log files.

6. Configure audit logging to specify how often you want to rotate the logs:

   lunash:> **audit config -parameter rotation -value** <value>

   For example, the command **audit config -parameter rotate -value hourly** would rotate the logs every hour, cutting down the size of individual log files, even in a situation of high-volume event recording, but would increase the number of files to be handled.

Log Entries

Log entries are made within the HSM, and are written to the currently active log file on the appliance file system. 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 on the appliance 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.

For SafeNet Luna Network HSM, to simplify configuration within its closed and hardened environment, the following rules apply:

- The maximum log file size is capped at 4 MB.
- The log path is internal to the SafeNet Luna Network HSM appliance.
- The rotation offset is set at 0.

Copying Log Files Off the Appliance

You can copy the log files off of the appliance for viewing and verification.

To copy files off the appliance

1. Create an archive of the logs that are ready to archive:

   lunash:> **audit log list**

   lunash:> **audit log tarlogs**

2. View a list of the log files currently saved on the appliance:

   lunash:>**my file list**

   For this example, assume that the list includes a file named **audit.tgz**.

3. On the computer where you wish to capture and store the log files, use **scp** (Linux) or **pscp** (Windows) to transfer the file from the appliance:

   /usr/safenet/lunaclient/logs :> **scp audit@myLunaHSM1:audit.tgz mylunsa1_audit_2014-02-28.tgz**

   Provide the audit user's credentials when prompted. This copies the identified file from the remote SafeNet Luna Network HSM's file system (in the **audit** account) and stores the copy on your local computer file system with a useful name.

4. You can view and parse the plain-text portion of the file.

5. You can verify the authenticity of the retrieved file using a connected HSM to which you have imported the Audit logging secret from the originating SafeNet Luna Network HSM.

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. On the SafeNet Luna Network HSM where HSM audit log files are being created, export the audit logging secret:

   lunash:> **audit secret export**

   The filename is displayed when the secret is exported. You can check the filename with **my file list**.

2. On a computer connected to both HSMs, use **scp** or **pscp** to transfer the logging secret from the appliance.

   - If you are planning to verify logs with a SafeNet Luna PCIe HSM, you can use the PCIe HSM's host computer.
   - If you are planning to verify logs with a second SafeNet Luna Network HSM, you must transfer the logging secret to a client computer, and then to the second appliance.

Linux	<client_install_dir>:> **scp audit@**<hostname_or_IP>**:**<log_secret_file> **.** Then, if transferring to a second SafeNet Luna Network HSM: <client_install_dir>:> **scp** <log_secret_file> **audit@**<hostname_or_IP>**:**
Windows	<client_install_dir>:> **pscp audit@** <hostname_or_IP>**:**<log_secret_file> **.** Then, if transferring to a second SafeNet Luna Network HSM: <client_install_dir>:> **pscp** <log_secret_file> **audit@**<hostname_or_IP>**:**

   This copies the identified file from the remote SafeNet Luna Network HSM's file system (in the "audit" account) and stores the copy on your local computer file system in the directory from which you issued the command. Provide the audit user's credentials when prompted.

3. 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**

4. 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>

5. 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.

Deciphering 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 bullet 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. If possible, use the remote logging feature to transmit log data to a Security Information and Event Management (SIEM) system to automatically analyze log data and identify anomalous events.
- Execute the **audit log tarlogs** LunaSH command regularly to archive the audit logs and transfer them to a separate machine for long term storage. Also, execute the **audit log clear** LunaSH command regularly to free up the audit log disk space on SafeNet Luna Network HSM.
- Consider installing and configuring a SafeNet Luna PCIe HSM in (or connected to) the remote log server to act as a "verification engine" for the remote log server. Ensure that the log secret for the operational HSM(s) has been shared with the log server verification HSM.

> **NOTE** This is not always possible, unless you are physically copying the logs over from the .tgz archive. Because log records do not necessarily appear on the remote log server immediately, the HMAC might be incorrect. Also, if more than one SafeNet Luna Network HSM is posting log records to a remote server, this could interfere with record counts.

- 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 callback service (**service restart cbs**).

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

> **CAUTION!** If the HSM is zeroized when a "disk full" condition has occurred, **hsm init** will fail, preventing the user from clearing the logs. This will effectively lock out the appliance and RMA may be necessary.

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

Remote Audit Logging

With SafeNet Luna Network HSM, the audit logs can be sent to one or more remote logging servers. Either UDP or TCP protocol can be specified. The default is UDP and port 514.

> **NOTE** You or your network administrator will need to adjust your firewall to pass this traffic (iptables).

UDP Considerations

If you are using the UDP protocol for logging, the following statements are required in the **/etc/rsyslog.conf** file:

```
$ModLoad imudp
$InputUDPServerRun (PORT)
```

Possible approaches include the following:

> With templates:

```
$template AuditFile,"/var/log/luna/audit_remote.log"
if $syslogfacility-text == 'local3' then ?AuditFile;AuditFormat
```

> Without templates:

```
local3.* /var/log/audit.log;AuditFormat
```

> Dynamic filename:

```
$template DynFile,"/var/log/luna/%HOSTNAME%.log"
if $syslogfacility-text == 'local3' then ?DynFile;AuditFormat
```

> **NOTE** The important thing to remember is that the incoming logs go to **local3**, and the port/protocol that is set on the SafeNet appliance must be the same that is set on the server running rsyslog.

Example using TCP

The following example illustrates how to setup a remote Linux system to receive the audit logs using TCP:

1. Register the remote Linux system IP address or hostname with the SafeNet Luna Network HSM:

   ```
   lunash:> audit remotehost add -host 192.20.9.160 -protocol tcp -port 1660
   ```

2. Modify the remote Linux system **/etc/rsyslog.conf** file to receive the audit logs:

   ```
   $ModLoad imtcp
   $InputTCPServerRun 514
   $template AuditFormat,"%msg:F,94:2%\n"
   #save log messages from SafeNet Luna Network HSM
   local3.* /var/log/luna/audit.log;AuditFormat
   ```

3. Modify the remote Linux system **/etc/sysconfig/rsyslog** file to receive the remote logs:

   ```
   # Enables logging from remote machines. The listener will listen to the specified port.
   SYSLOGD_OPTIONS="-r -m 0"
   ```

4. Restart the rsyslog daemon on the remote Linux system:

   ```
   # service rsyslog restart
   ```

5. Monitor the audit logs on the remote Linux system:

   ```
   # tail -f /var/log/luna/audit.log
   ```

CHAPTER 2:
Backup and Restore HSMs and Partitions

SafeNet Luna Network HSMs secure the creation, storage, and use of cryptographic data (keys and other objects). However, no device can protect completely against unforeseen damage from various sources, including disaster-scale events. Therefore, the SafeNet Luna HSM product line provides several ways to protect secure copies of your important objects and keys at safe locations and to later restore your important data to your production, or primary HSM, in case of need.

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

Backup and Restore Overview and Best Practices

This section provides an overview of the various ways you can backup and restore your HSM partitions, and provides some guidance for best practices to ensure that your sensitive key material is protected in the event of a failure or other catastrophic event. It contains the following topics:

Backup and Restore Best Practices

To ensure that your data is protected in the event of a failure or other catastrophic event, Thales recommends that you use the following best practices as part of a comprehensive backup strategy:

> Develop and document a backup and recovery plan. This plan should include the following:
- What is being backed up
- The backup frequency
- Where the backups are stored
- Who is able to perform backup and restore operations
- Frequency of exercising the recovery test plan

> Make multiple backups. To ensure that your backups are always available, build redundancy into your backup procedures.

> Use off-site storage. In the event of a local catastrophe, such as a flood or fire, you might lose both your working HSMs and locally stored backup HSMs. To fully protect against such events, always store a copy of your backups at a remote location. You can automate off-site backups using the remote backup feature, See "Backup and Restore From the Client to a Remote Backup HSM (LunaCM, RBS)" on page 63 for more information.

> Regularly exercise your disaster recovery plan. Execute your recovery plan at least semi-annually (every six months) to ensure that you can fully recover your key material. This involves retrieving your stored Backup HSMs and restoring their contents to a test partition, to ensure that the data is intact and that your recovery plan works as documented.

****WARNING** Failure to develop and exercise a comprehensive backup and recovery plan may prevent you from being able to recover from a catastrophic event. Although Thales provides a robust set of backup hardware and utilities, we cannot guarantee the integrity of your backed-up key material, especially if stored for long periods. Thales strongly recommends that you exercise your recovery plan at least semi-annually (every six months) to ensure that you can fully recover your key material.**

Backup and Restore Options

The available options for backing up your SafeNet Luna Network HSM partitions include:

> Local or remote backup to a SafeNet Luna Backup HSM (see "Local Partition Backup and Restore Using the Backup HSM" on page 1 and "Backup and Restore From the Client to a Remote Backup HSM (LunaCM, RBS)" on page 63)

> Key synchronization among two or more SafeNet Luna HSMs in an HA configuration (see "High-Availability (HA) Configuration and Operation" on page 128)

> Any combination of the above methods, to suit your needs

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

How Partition Backup Works

HSM partition backup securely clones partition objects from a named HSM partition, to a SafeNet Luna Backup HSM (supports remote or local backups). This allows you to safely and securely preserve important keys, certificates, etc., away from the primary SafeNet Luna HSM. 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.

To back up a partition, you must own it and be able to see it. You can use LunaSH to back up any partitions you own on a SafeNet Luna Network HSM appliance, or LunaCM to backup any SafeNet Luna Network HSM partitions that are visible as slots.

When you backup a partition, the contents of your HSM partition are copied to a matching partition on the SafeNet Luna Backup HSM. You can add to, or replace, objects in the backup archive, as follows:

- Partition backups initiated with the **add** or **append** option add new or changed objects to the partition archive, leaving existing objects intact.
- Partition backups initiated with the **replace** option replace all existing objects in the partition archive with current contents of the partition, destroying the existing objects.

The backup operation can go from a source partition on a SafeNet Luna HSM to an existing partition on the Backup HSM, or if one does not exist, a new partition can be created during the backup. The restore operation, however, cannot create a target partition on a SafeNet Luna HSM; it must already exist.

You can restore a partition backup to the original source HSM or to a different SafeNet Luna HSM. The HSM you restore to must already have a suitable partition created for the restored objects. The partition can have any name - it does not need to match the name of the archive partition on the backup device.

Backup Devices

You can back up all of your partitions to a SafeNet Luna Backup HSM:

SafeNet Luna Backup HSM (Backup HSM)

> **NOTE** The word "Remote" in the product name merely indicates that the SafeNet Luna Backup HSM provides remote backup capability. It also supports local backup and restore. The SafeNet Luna Backup HSM is commonly referred to as the Backup HSM.

The SafeNet Luna Backup HSM (Backup HSM) is a separately powered unit that you can connect as follows:

- To the USB port of a a SafeNet Luna Network HSM appliance. This allows a SafeNet Luna Network HSM administrator to use LunaSH to back up any partitions on the appliance that they own (non-PSO partitions).
- To the USB port of a local SafeNet Luna HSM client workstation. This allows the workstation administrator to use LunaCM to back up any SafeNet Luna PCIe HSM devices installed in the workstation or any SafeNet Luna Network HSM partitions registered to the workstation.
- To the USB port of a remote SafeNet Luna HSM client workstation running the Remote Backup Service (RBS). You can then register the Remote Backup HSM with a local SafeNet Luna HSM client workstation so that the it sees the Remote Backup HSM as a slot in LunaCM. This allows the administrator of the local SafeNet Luna HSM client workstation to use LunaCM to back up any local slots to the remote Backup HSM.

Performing a Backup

To perform a backup, you identify the partition to be backed up (source), and the partition that will be created (or added to) on the Backup HSM. You can specify whether to **add/append** only unique objects (objects that have not previously been saved onto the target partition), or to **replace** (overwrite) the objects on the target partition.

LunaSH

If you are using LunaSH to backup a partition on a SafeNet Luna Network HSM, use:

partition backup -partition <partition_label> **-tokenpar** <backup_label> **-serial** <backup_HSM_SN> [**-add**] [**-replace**]

More options are available. See "partition backup" on page 1 in the *LunaSH Command Reference Guide* for full command syntax.

LunaCM

If you are using LunaCM on a Client workstation, first login to the partition as Crypto Officer.If the backup device is

- a slot in the current system, use:

  partition archive backup -slot <backup_slot> **-partition** <name_for_backup> [**-append**] [**-replace**]

- in a remote workstation, use:

  partition archive backup -slot remote-hostname <hostname> **-port** <portnumber> **-partition** <name_for_backup> [**-append**] [**-replace**]

- a USB-attached HSM, use:

  partition archive backup -slot direct -partition <backup_partition> [**-append**] [**-replace**]

More options are available. See "partition archive backup" on page 1 in the *LunaCM Command Reference Guide* for full command syntax.

LunaCM assumes that the target partition already exists with the appropriate domain, while LunaSH expects you to provide the domain, or prompts you if it is not provided (for password-authenticated HSMs).

Replacing or Appending

If a matching target partition exists and the source partition is being incrementally backed up, choosing the **add/append** 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. The existing partition is erased and a new one created.

Objects are Smaller When Stored on Backup HSM

Objects stored on the Backup HSM may be smaller than the same objects stored on the SafeNet Luna Network HSM. For example, symmetric keys are 8 bytes smaller when stored on the Backup HSM. This size difference has no effect on backup and restore operations.

Comparison of Backup Performance by Medium

For reference, this table shows examples of time required for a backup operation for one partition containing 25 RSA 2048-bit keypairs, or 50 objects in total. The source is a SafeNet Luna Network HSM appliance. The destination backup devices and paths are listed in the table.

Backup Destination	Time Required for Operation	Comment
SafeNet Luna Backup HSM (PW-auth), local	5 seconds	Password is supplied with the command
SafeNet Luna Backup HSM (PED-auth), local	5 seconds plus...	Add any time required for PED key operations

Compatibility with Other Devices

Backup can co-exist with PKI Bundle operation. That is, multiple devices can be connected simultaneously to a SafeNet appliance (three USB connectors). Thus, you could connect a SafeNet Luna Backup HSM, a SafeNet DOCK 2 (with migration-source tokens in its reader slots), and a SafeNet Luna USB HSM to the three available USB connectors on the SafeNet Luna Network HSM.

Why is Backup Optional?

In general, a SafeNet Luna HSM or HSM partition is capable of being backed up to a SafeNet Luna Backup HSM. 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 devices are an optional equipment for SafeNet Luna 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 cert. 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 keypair, for safe and legal backup.

How Long Does Data Last?

SafeNet Luna 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 critical key material, and the firmware that makes the device work.

No electronic storage is forever. If your SafeNet Luna 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.

Additional Operational Questions

Is SafeNet Luna Backup HSM capable of backing up multiple SafeNet Luna HSMs or is it a one-to-one relationship?
For example, if we had two SafeNet Luna Network HSM appliances each with two partitions, or if we had four SafeNet Luna PCIe HSMs, could we backup all four partitions to a single Backup HSM? If yes, do they need to be under the same domain?

Answer

One SafeNet Luna Backup HSM can back up multiple SafeNet Luna HSMs. The domains on those SafeNet Luna HSMs do not need to match each other (although they can, if desired), since domains can be partition-specific. The only domains that must match are those on any given SafeNet Luna HSM partition and its backup partition on the SafeNet Luna Backup HSM. With that said, the limits on quantity of backup of partitions from multiple appliances or embedded HSMs is the remaining space available on the Backup HSM, and the remaining number of partitions (base configuration for SafeNet Luna Backup HSM is 20 partitions - you can purchase additional capability).

Can a SafeNet Luna Backup HSM keep multiple backups of a single partition?
For example, could we perform a backup of an application partition one month and then back it up again next month without overwriting the previous month?

Answer

Yes, you can do this as long as each successive backup partition (target) is given a unique name.

About the SafeNet Luna Backup HSM

This section describes what you can do with the SafeNet Luna Backup HSM (Backup HSM) and outlines the various ways, both local and remote, that you can connect the Backup HSM to perform backup and restore operations. It contains the following topics:

> "Functionality of the SafeNet Luna Backup HSM" on the next page

> "Backup and Restore Options and Configurations" on page 46

> **NOTE** The word "Remote" in the product name merely indicates that the Backup HSM provides remote backup capability. You can use the SafeNet Luna Backup HSM to back up the contents of your HSM to a locally attached Backup HSM, or to a remotely located Backup HSM. The SafeNet Luna Backup HSM is referred to as the Backup HSM in this section.

Functionality of the SafeNet Luna Backup HSM

You can use the SafeNet Luna Backup HSM to backup multiple partitions from one or more SafeNet Luna Network HSMs or SafeNet Luna PCIe HSMs. Partition domain and authentication attributes are maintained when you back up a partition, which impacts how you can use the Backup HSM.

Storage Capacity and Supported Number of Partitions

Backup is performed on a per-partition basis. SafeNet Luna PCIe HSM supports one application partition. The SafeNet Luna Network HSM supports multiple application partitions. The size of a SafeNet Luna Network HSM partition is configurable, but since all partitions share the HSM memory, the more partitions you create, the smaller they must be.

The base configuration for SafeNet Luna Backup HSM is 20 partitions and 15.5 Mb of space, allowing you to backup a SafeNet Luna Network HSM with up to twenty partitions, or any combination of partitions on individual SafeNet Luna Network HSMs, up to the maximum memory available on the Backup HSM. SafeNet Luna Network HSMs can be updated to support up to 100 partitions. You have the option of purchasing and adding capability upgrades for 50 or 100 partitions to SafeNet Luna Network HSM, as well as to the SafeNet Luna Backup HSM.

> **NOTE** The size of the partition header is different for a SafeNet Luna Network HSM partition and its equivalent backup partition stored on a SafeNet Luna Backup HSM. As a result, the value displayed in the Used column in the output of the **partition list** command (for the backed up SafeNet Luna Network HSM partition) is different than the value displayed in the Used column in the output of the **token backup partition list** command (for the backup partition on the Backup HSM).

Upgrading the Number of Supported Partitions

When your SafeNet Luna Backup HSM is connected locally to a SafeNet Luna Network HSM appliance, use the upgrade instructions at "HSM Capability and Partition Upgrades" on page 310 to apply an upgrade to increase the number of HSM partitions that can be backed up to the device.

Domains and Backups

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 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.

PED or Password Authentication

The Backup HSM creates a partition with matching authentication type to the SafeNet Luna Network HSM partition that is being backed up. That does not work in the opposite direction, however. The Backup HSM 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 Backup HSMs 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). Normally this is not a concern because a given installation is likely to employ all SafeNet Luna Network HSMs of the same authentication type.

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.

Backup and Restore Options and Configurations

The SafeNet Luna Backup HSM supports local or remote HSM backup. The options for backup of primary/source SafeNet Luna Network HSMs are:

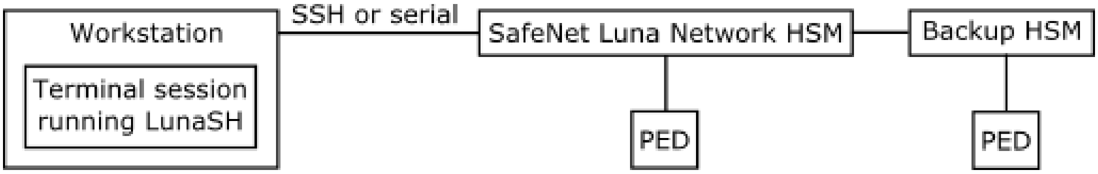
- **Local backup of any SafeNet Luna Network HSM**, where all components are co-located. This is a possible scenario with all HSMs, but is more likely with direct-connect, local-to-the-client HSMs such as SafeNet Luna PCIe HSM. It is unlikely for SafeNet Luna Network HSMs, simply because SafeNet Luna Network HSMs normally reside in a server rack, distant from its administrators.
- **Local backup of SafeNet Luna Network HSM**, where the 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 the 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 Luna Network HSM**, where the SafeNet Luna Network 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 must connect (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 the Backup HSM directly, via USB, to a SafeNet Luna Network HSM appliance or SafeNet Luna PCIe HSM host server. That is, local backup is local to the HSM being backed-up, not necessarily local to the administrator who is directing the process, who might be far away.

For remote backup, you connect the Backup HSM 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 HSMs, you must have a copy of the appropriate red (domain) PED keys to use with the Backup HSM in order to perform the copy/cloning (backup and restore) operation between the HSMs.

Backing Up a Local HSM to a Directly Connected Backup HSM

The simplest way to backup your SafeNet Luna Network HSM is to connect the Backup HSM directly to the SafeNet Luna Network HSM appliance. To perform a backup/restore, you open an SSH or serial connection from your workstation to the appliance, and then launch LunaSH in a terminal session to perform the backup, as illustrated in the following figure:

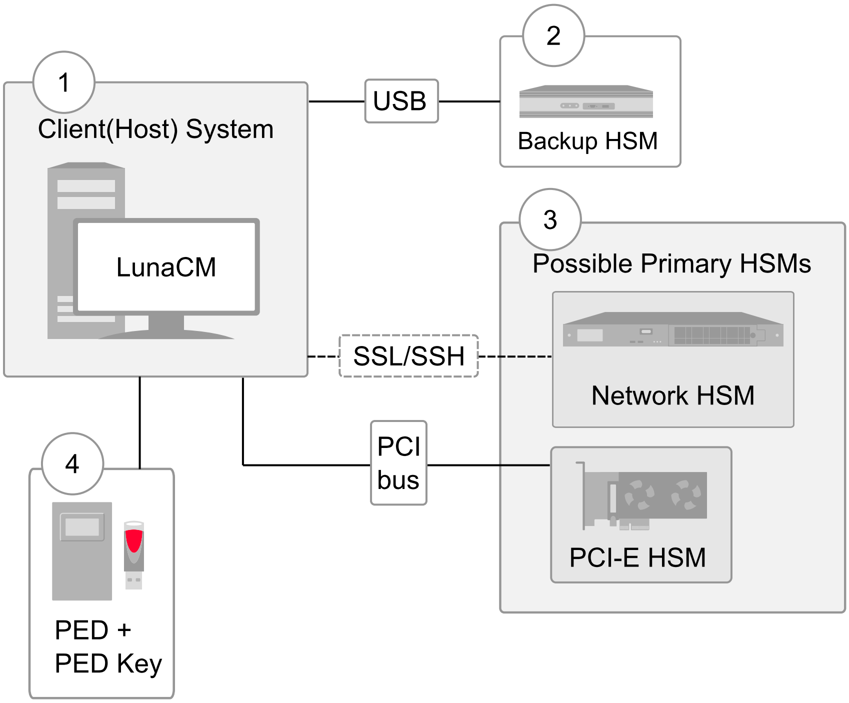


The workstation is simply a display terminal for LunaSH running on the appliance. It does not require the SafeNet Luna Client software.

The PEDs are required only if the SafeNet Luna Network HSM is PED-authenticated. The appropriate SO (blue), partition (black) and domain (red) PED keys are required.

Backup to a Backup HSM Connected to a Local Client

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



1	LunaCM on the client (host) system sees the primary and backup slots and controls the backup/restore operation.
2	Backup HSM is a slot visible to the client (host) system when it runs LunaCM.
3	Working HSMs are slots visible to the client (host) system when it runs LunaCM.
4	Every slot on the backup must have same domain (red PED key) as matching slot on the primary HSMs.

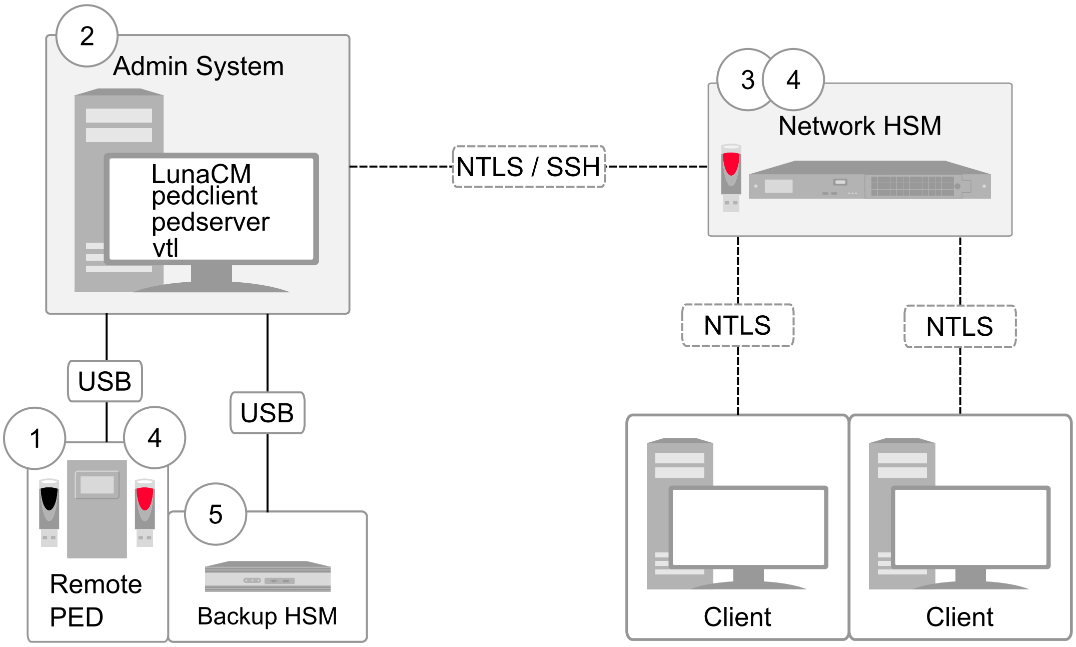
The other two backup and restore options, local backup of a distant SafeNet Luna Network HSM and remote backup of any SafeNet Luna Network HSM require that PED operations be performed remotely. For that reason, HSMs must be prepared (locally) in advance by having orange Remote PED keys created and matched with each HSM.

Backing Up a Remote HSM to a Locally-Connected Backup HSM

The diagram below summarizes the elements and setup for backing up partitions of a remote SafeNet Luna Network HSM to a Backup HSM that is attached to the local host. For this example, the system administrator (admin) for the SafeNet Luna Network HSM appliance is also the person doing the backup. The local host is configured as follows:

- The SafeNet Luna HSM client software with the Remote PED options is installed.
- A Remote Luna PED is connected.
- The SafeNet Luna Backup HSM is connected.

Before performing a backup, the admin must open an SSH session to the SafeNet Luna Network HSM appliance and perform a certificate exchange and registration for each SafeNet Luna Network HSM partition to be backed up to make the local host a client of the partitions.



1	The admin must have client access to each partition being backed up. In this scenario, the admin must have black PED keys and passwords for the partitions.
2	The local host is used to control the backup/restore. The SafeNet Luna Network HSM client vtl software is used to generate and trade certificates with SafeNet Luna Network HSM, to create an NTLS link. The Luna PEDServer software running on the local host, in conjunction with the PEDClient software running on the SafeNet Luna Network HSM, provides remote PED access to the SafeNet Luna Network HSM.

3 The local host can see the SafeNet Luna Network HSM partitions as slots in LunaCM. The Luna PEDClient software runs on the SafeNet Luna Network HSM when it needs to access the Remote PED via the Luna PEDServer software running on the local host.

4 Every slot on the Backup HSM must have same domain (red PED key) as the matching slot on the working HSM. The domain (red) PED keys can be different for each partition or they can share one common domain, re-used for all partitions. The important consideration is that whatever domain situation exists on the primary HSM must be matched on the Backup HSM.

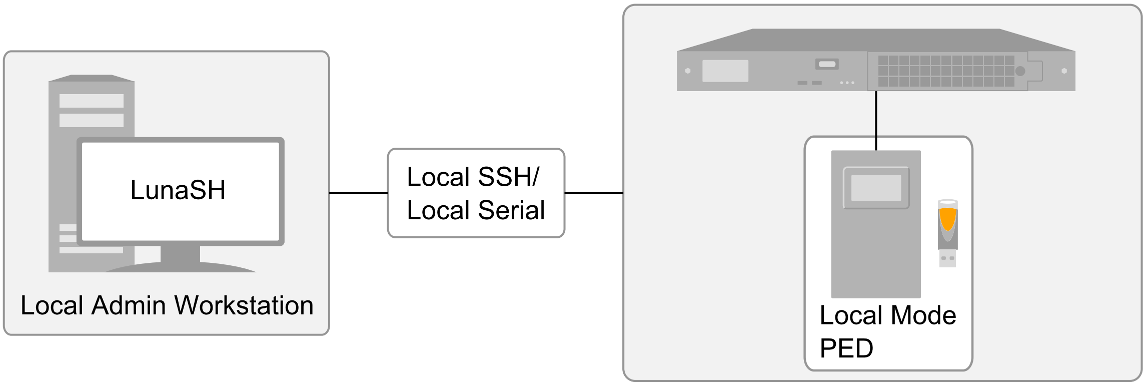
5 The local host can see the Backup HSM as a slot in LunaCM. Because the local host views the backup/restore operation in this scenario as a local transaction, between two slots visible to LunaCM on the local host, the remote backup service (RBS) is not needed.

This scenario avoids the complication of an intermediary computer (as would be needed for true remote backup), but at the cost of giving the authentication keys for all client partitions to an administrator. Your security protocol determines whether this is acceptable.

Backing Up a Remote HSM to a Remotely-Connected Backup HSM

This section describes how to backup a remote HSM to a Backup HSM that is connected over the network to a remote host. In this configuration, you require an orange PED key, imprinted with the Remote PED Vector (RPV) for the HSM you want to back up. To create the orange PED key, you must temporarily connect a PED directly to the HSM you want to back up, as illustrated in the following figure. The figure shows a local admin session to the HSM. You could administer remotely, but this operation nevertheless requires a local PED connection to the HSM and someone there to insert PED keys and press buttons on the PED keypad, so we depict the most likely connection situation - one person doing all jobs at one location. 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 shipped to its distant location. See "Initializing the Remote PED Vector (RPV) and Creating the Orange PED Key" on page 209.

Figure 1: Creating an orange PED key imprinted with the remote PED vector (RPV) for the HSM



After you have created the orange (RPV) PED key and have the appropriate red (domain) PED keys for the partitions you want to back up, you are ready to configure and use your Remote Backup HSM. In this scenario, you could have as many as three different computers (we depict two for our example) connecting to the SafeNet Luna Network HSM:

- one to run the ssh administrative connection to the shell (lunash:>) on the SafeNet Luna Network HSM appliance

- one to run the Remote PED server, with the Luna PED (in remote mode) connected via USB to the computer and separately connected to the mains electrical power source (see "Changing Modes" on page 202 for instructions on changing modes on the Luna PED)
- one to run a client session with vtl and the SafeNet Remote Backup driver, and with the SafeNet Luna Backup HSM with its own local Luna PED attached

As noted previously, the orange PED keys contain a Remote PED Vector (RPV) that matches the RPV inside the SafeNet Luna Network 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. At the same time, the SafeNet Luna Network HSM and the SafeNet Luna Backup HSM must share the same cloning domain, in order for backup and restore (cloning) operations to take place between the two HSMs. Therefore, red PED keys with that cloning domain must be available.

SafeNet Luna Network HSMs use Remote Backup Service (RBS) to facilitate Remote Backup.

Required Software

LunaCM is required on both the Client (Host) System and on the System Admin computer, but is run on Client (Host) System to launch and manage the backup and restore activity. PEDClient is needed on both the Client (Host) System and the System Admin computer, as well as on any SafeNet Luna Network HSM.

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

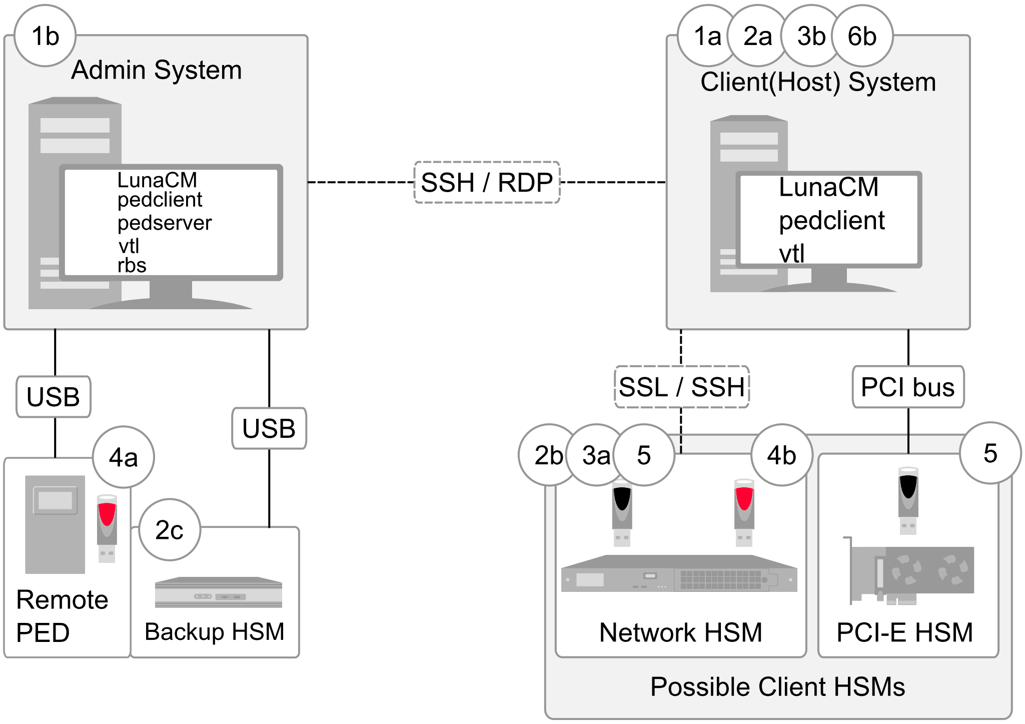
PEDserver must reside (and run, waiting for calls) on any computer connected to a Remote PED.

RBS is required on the computer connected to the SafeNet Luna Backup HSM. RBS is not needed on any other computer in the scenario.

Example

The following figure provides an example configuration for backing up a remote HSM to a backup HSM connected to a remote host. This scenario adds an intermediate computer (Client (Host) System) to broker the remote backup of the HSM partitions. That could be a special-purpose computer, or it could simply mean that the Admin on the computer with the Remote Backup HSM is given remote access to each client that normally uses a SafeNet Luna Network HSM partition. The tradeoff is that those clients already have access to their registered partitions, so there is no need for the Remote Backup HSM admin to have client access (PED keys) for those partitions. Your security protocol dictates which scenario is appropriate for you.

Figure 2: Configuration for backing up a remote HSM to a backup HSM connected to a remote host



1	"Client (Host) System" (1a) is a client of the SafeNet Luna Network HSM being backed up, but "System Admin" (1b) is not a client of SafeNet Luna Network HSM.
2	LunaCM on "Client (Host) System" (2a) sees the primary (2b) and backup (2c) slots and controls the backup/restore.
3	Each SafeNet Luna Network HSM (3a) partition is a slot visible to a "Client (Host) System" (3b) when Client (Host) System runs LunaCM.
4	Every slot on the backup (4a) must have same domain (red PED key) as 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	Backup HSM (6a) is a slot visible to "Client (Host) System" (6b) when Client (Host) System runs LunaCM.

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" on the next page

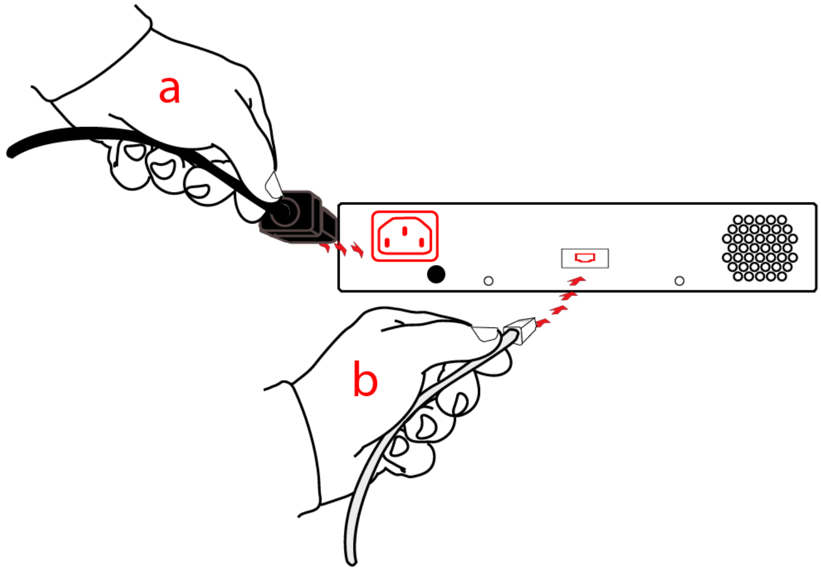
- "Disconnecting a Backup HSM" on the next page
- "Installing the Battery" on the next page
- "Backup HSM Storage and Maintenance" on page 56

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

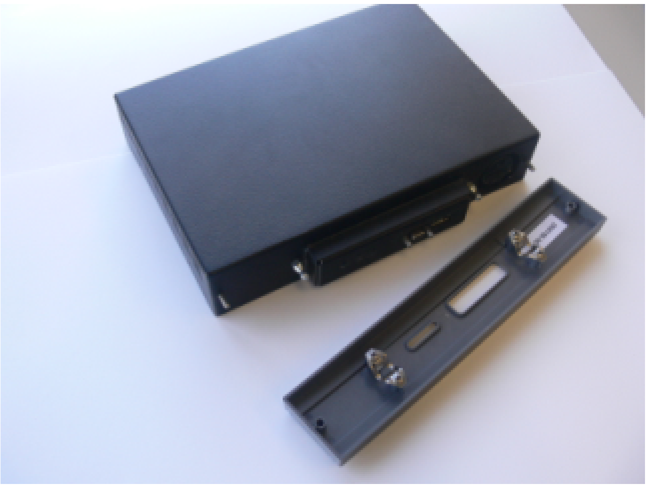
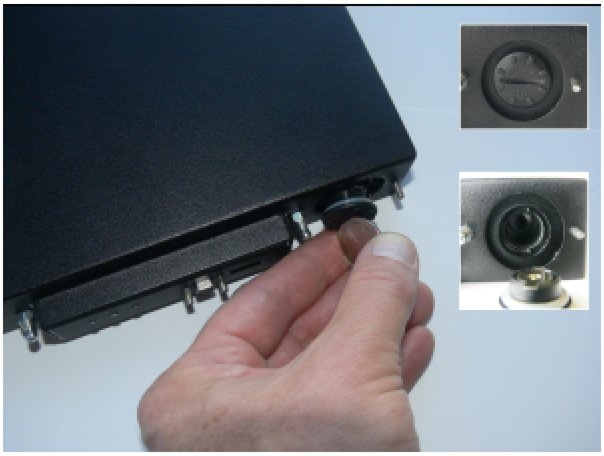
The Backup HSM is a USB device. It is not equipped with a power switch. There is no special procedure for disconnecting or shutting down a SafeNet Luna Backup HSM.

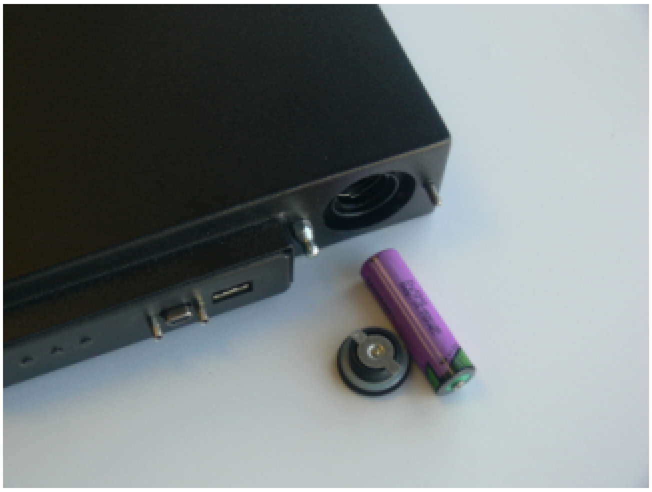
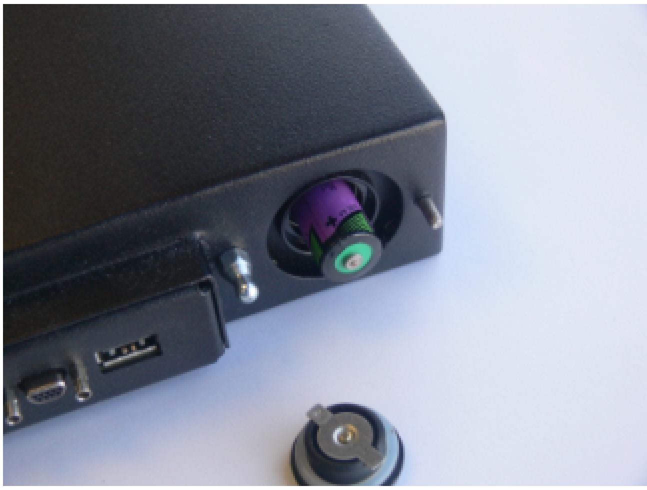
If the Backup HSM is used in remote configuration for SafeNet Luna Network HSM (connected to a workstation acting as backup server), then your only action is to do the usual dismount of a USB device (for the benefit of your workstation, not the Backup HSM - “It is now safe to disconnect your USB Device”).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

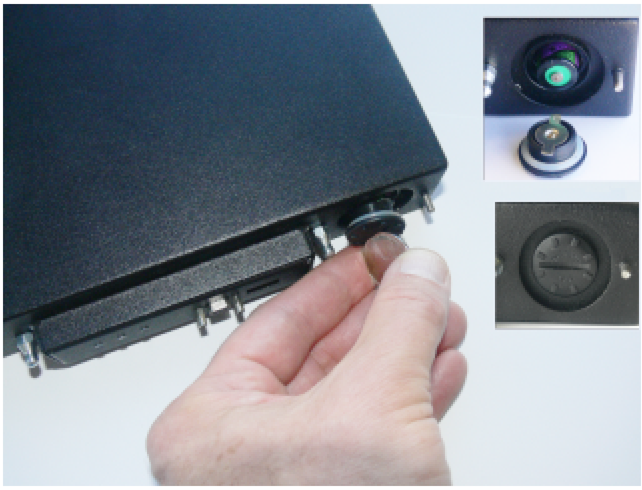
If the Backup HSM is connected to SafeNet Luna Network HSM for local backup, you have no access to the SafeNet Luna Network HSM’s internal hardened kernel, so you cannot issue an un-mount instruction. Simply disconnect the cables and the system figures it out at either end. Both SafeNet Luna Network 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 and Restore From the Client to a Local Backup HSM (LunaCM)

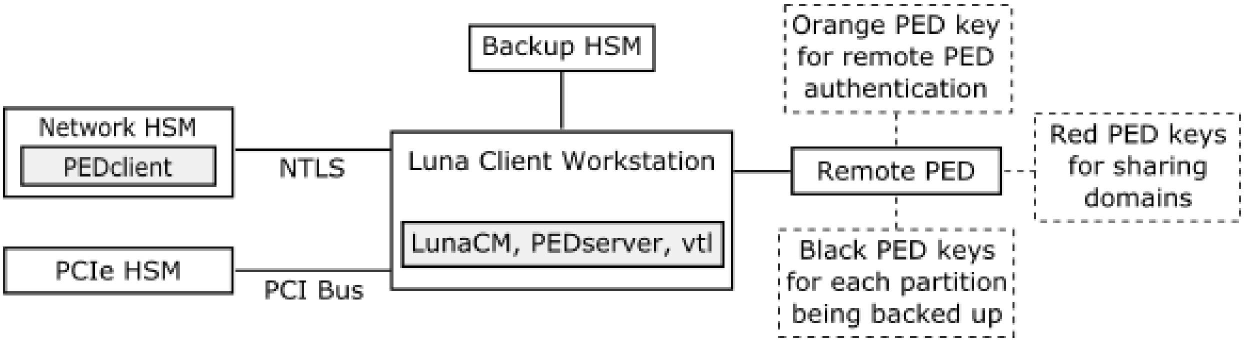
This section describes how to use LunaCM to backup and restore a partition from the client to a locally connected SafeNet Luna Backup HSM (Backup HSM). To perform a local backup, you connect the SafeNet Luna Backup HSM to a USB port on the SafeNet Luna HSM client workstation and use LunaCM to log in as the Crytpo Officer (CO) and backup any SafeNet Luna Network HSM or SafeNet Luna PCIe HSM partitions that are visible as slots.

The backup operation can go from a source partition (on a SafeNet Luna Network HSM) to an existing partition on the Backup HSM, or if one does not exist, a new partition can be created during the backup. The restore operation, however, cannot create a target partition on a SafeNet Luna Network HSM; it must already exist.

You can restore a partition backup to the source HSM or to a different SafeNet Luna Network HSM. The HSM you restore to must already have a suitable partition created for the restored objects. The partition can have any name - it does not need to match the name of the source partition on the backup HSM.

You can connect the Backup HSM to a SafeNet Luna HSM client workstation to backup any SafeNet Luna Network HSM or SafeNet Luna PCIe HSM partitions that are visible as slots in LunaCM, as illustrated in the following figure:

Figure 3: Configuration for SafeNet Luna Network HSM/PCIe partition backup/restore using a Backup HSM connected to a local client workstation


Backup HSM
Orange PED key for remote PED authentication
Network HSM
PEDclient
NTLS
Luna Client Workstation
Remote PED
Red PED keys for sharing domains
LunaCM, PEDserver, vtl
PCIe HSM
PCI Bus
Black PED keys for each partition being backed up

In this configuration, you connect the Backup HSM and SafeNet Remote PED, via USB, to your SafeNet Luna HSM client workstation. The SafeNet Luna Network HSM appliance is remote to the SafeNet Luna HSM client workstation and is connected using NTLS. Any installed PCIe devices communicate with the SafeNet Luna HSM client over the PCI bus.

Any partitions you want to backup must be registered with the SafeNet Luna HSM client workstation, and be visible as slots in LunaCM. The Backup HSM most also be visible as a slot.

If you are backing up PED-authenticated partitions, you require a PED. If you want to backup SafeNet Luna Network HSM partitions, the PED must have remote capability (Remote PED). Remote PED uses the pedserver/pedclient processes running on the SafeNet Luna HSM client workstation and on the SafeNet Luna Network HSM appliance to provide remote PED services for the network-attached SafeNet Luna Network HSM appliance. The PED provides authentication for all of the attached HSMs (the USB-connected SafeNet Luna Backup HSM, the NTLS-connected SafeNet Luna Network HSM, and the PCI bus-connected SafeNet Luna PCIe HSM). Every slot on the backup must have same domain (red PED key) as the matching slot on the source HSMs.

> **NOTE** 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 HSM. 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.

Backing Up a Partition to a Locally Connected Backup HSM

You can backup any slots you can see on the client workstation. You must log in as the Crypto Officer to the partition you want to backup.

To backup an application partition to a Backup HSM connected to a SafeNet Luna HSM client workstation:

1. Configure the remote PED, as described in "Remote PED Setup" on page 209.
2. Start the LunaCM utility on the SafeNet Luna HSM client workstation.

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

LunaCM V7.0 - Copyright (c) 2006-2017 Gemalto, Inc.

        Available HSM's:

        Slot Id ->              1
        HSM Label ->            SA52_P1
        HSM Serial Number ->    500409014
        HSM Model ->            LunaSA
        HSM Firmware Version -> 7.0.1
        HSM Configuration ->    Luna User Partition With SO (PED) Signing With Cloning Mode
        HSM Status ->           OK

        Slot Id ->              2
        HSM Label ->            BackupHSM Serial Number ->    700101
        HSM Model ->            G5Backup
        HSM Firmware Version -> 6.26.0
        HSM Configuration ->    Remote Backup HSM (PED) Backup Device
        HSM Status ->           OK

        Current Slot Id: 1
```

3. Use the **slot set** command to go to the slot you want to back up:

```
lunacm:> slot set slot 1

        Current Slot Id: 1     (Luna User Slot 7.0.1 (PED) Signing With Cloning Mode)

Command Result : No Error
```

4. Establish that the HSM is listening for a SafeNet Remote PED:

```
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
```

The SafeNet Luna Network HSM is now listening for PED interaction via the link between PedClient on the SafeNet Luna Network HSM appliance and PedServer on the workstation, and is not expecting a PED connected directly at the location of the SafeNet Luna Network HSM.

5. Log in as the Crypto Officer (CO) to the partition in the current slot. This is the partition that you want to back up:

```
lunacm:> role login -name Crypto Officer

        Option -password was not supplied.  It is required.

        Enter the password: *******

        User is activated, PED is not required.

Command Result : No Error
```

6. Disconnect the PED from your source HSM (slot 1 in this example), and connect to the Backup HSM (slot 2 in this example). The PED remains physically connected by USB cable to the SafeNet Luna HSM client workstation, and remains in Remote mode - you are merely changing slots that are in conversation with that PED.
   a. First, tell the SafeNet Luna Network HSM to disconnect from Remote PED with the command **ped disconnect**.
   b. Tell the Backup HSM to connect to Remote PED (it makes no difference that the PED and the Remote Backup HSM are USB-connected to the same workstation/laptop; when use of Remote PED is invoked by command **ped connect** and verified by **ped get**, all HSM-PED interaction takes place between PedClient running on that workstation and PedServer, also running on that workstation).

```
lunacm:> ped connect ip 192.20.10.189 -slot 2

Command Result : No Error

lunacm:> ped get -slot 2

        HSM slot 2 listening to remote PED (PED id=100).

Command Result : No Error
```

7. Use the **partition archive backup** command to 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 archive backup -slot 2 -par SAbck1

        Logging in as the SO on slot 2.
        Please attend to the PED.

        Creating partition SAbck1 on slot 2.
        Please attend to the PED.

        Logging into the container SAbck1 on slot 2 as the user.
        Please attend to the PED.

        Creating Domain for the partition SAbck1 on slot 2.
        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).

        Backup Complete.

        20 objects have been backed up to partition SAbck1
        on slot 2.

Command Result : No Error
```

8. Backup is complete, and can be verified if you like.

Restoring a Partition from a Locally Connected Backup HSM

You can restore a backup to any slot you can see on the client workstation. You must log in as the Crypto Officer to the partition you want to restore to.

To restore an application partition from a Backup HSM connected to a SafeNet Luna HSM client workstation:

1. Create a target partition for the restore operation on the HSM you are restoring to, if it does not already exist, and register the partition with the SafeNet Luna HSM client workstation so that it is visible as a slot in LunaCM.
2. Start the LunaCM utility on the SafeNet Luna HSM client workstation.

```
LunaCM v7.0.0. Copyright (c) 2006-2017 SafeNet.

        Available HSMs:

        Slot Id ->              0
        Label ->                par1
        Serial Number ->        154438865288
        Model ->                LunaSA 7.0.0
        Firmware Version ->     7.0.1
        Configuration ->        Luna User Partition With SO (PED) Signing With Cloning Mode
        Slot Description ->     Net Token Slot

        Slot Id ->              21
        Label ->                lunabackup
        Serial Number ->        496771
        Model ->                G5Backup
        Firmware Version ->     6.26.0
        HSM Configuration ->    Remote Backup HSM (PED) Backup Device
        HSM Status ->           OK


        Current Slot Id: 0
```

3. Use the **slot set** command to go to the slot you want to restore to.

```
lunacm:> slot set slot 0

        Current Slot Id: 0     (Luna User Slot 7.0.1 (PED) Signing With Cloning Mode)

Command Result : No Error
```

4. Open a remote PED session to the SafeNet Luna Network HSM you are restoring to:

```
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
```

The SafeNet Luna Network HSM is now listening for PED interaction via the link between PEDclient on the SafeNet Luna Network HSM appliance and PEDserver on the workstation, and is not expecting a PED connected directly at the location of the SafeNet Luna Network HSM.

5. Log into the partition in the current slot. This is the partition that you want to restore to.

```
lunacm:> role login -name Crypto Officer

        Option -password was not supplied.  It is required.

        Enter the password: *******

        User is activated, PED is not required.

Command Result : No Error
```

6. Use the **partition archive restore** command restore the partition from the Backup HSM to the current slot, adding to, or replacing, the current partition contents:

```
partition archive restore -slot <backup-hsm-slotnumber> -partition LunaSAPartitionname -
password ClientPassword -replace
```

> **NOTE** In the command above, you can use **-add** instead of **-replace**. Adding might 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. The two would be assigned different handles, however.

Backup and Restore From the Client to a Remote Backup HSM (LunaCM, RBS)

This section describes how to use LunaCM and the Remote Backup Service (RBS) to backup and restore a partition from the client to a remotely located SafeNet Luna Backup HSM (Backup HSM). It contains the following sections:

- "Overview" below
- "Configuring the Remote Backup Service (RBS)" on the next page
- "Backing Up an Application Partition to a Remotely Located Backup HSM" on page 67
- "Restoring an HSM Partition From a Remotely Located Backup HSM" on page 71

Overview

Remote backups are enabled by the SafeNet Remote Backup Service (RBS). RBS is a utility, included with the SafeNet Luna HSM client software, that runs as a service (Windows) or daemon (Unix/Linux) on a workstation used to host one or more remote Backup HSMs.

To use RBS, do the following:

1. Configure it to define which of the Backup HSMs connected to the workstation running RBS that you want to make available to other SafeNet Luna HSM client workstations or SafeNet Luna Network HSM appliances for performing remote backups.
2. Register the workstation running RBS with any SafeNet Luna HSM client workstations or SafeNet Luna Network HSM appliances that you want to be able to use the Remote Backup HSMs.
3. Start the RBS service/daemon.

Once RBS is configured and running, the SafeNet Luna HSM client workstations or SafeNet Luna Network HSM appliances registered with the workstation running RBS can see its available Backup HSMs as slots in LunaCM (SafeNet Luna HSM client workstation) or LunaSH (SafeNet Luna Network HSM appliance). To perform backup and restore operations using the Remote Backup HSMs, you open a LunaCM or LunaSH session, as relevant, on the SafeNet Luna HSM client workstation or SafeNet Luna Network HSM appliance used to host the slot you want to backup, and specify the slot for the Remote Backup HSM as the slot to use for the backup/restore operation.

The backup operation can go from a source partition (on a SafeNet Luna 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. The restore operation cannot create a target partition on a SafeNet Luna Network HSM; it must already exist and have a registered NTLS link.

To back up PED-authenticated partitions, you can connect a remote PED to the Backup HSM host workstation, or you can use a separate computer to provide PED operations.

NOTE Remote PED (PED Server) is supported on Windows only.

Configurations for Remote Backup of a SafeNet Luna Client Workstation Slot

The possible configurations for performing a remote backup of a SafeNet Luna HSM client workstation slot are illustrated in the following figures. Only PED-authenticated backup configurations are shown.

Figure 4: Configuration for remote backup of a SafeNet Luna HSM client workstation slot with the remote PED connected to the backup workstation

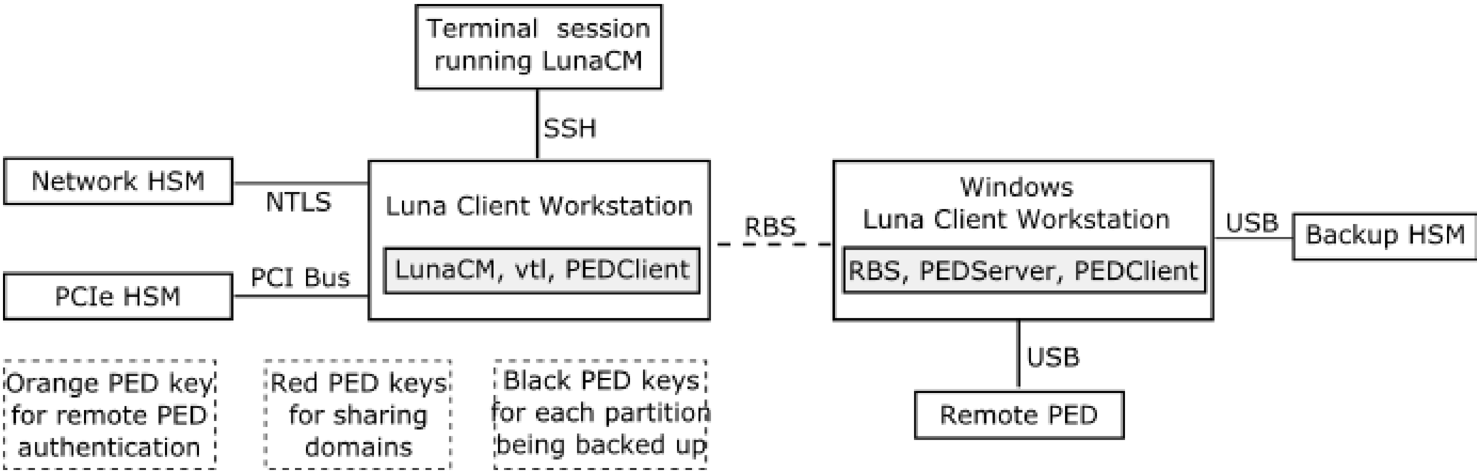
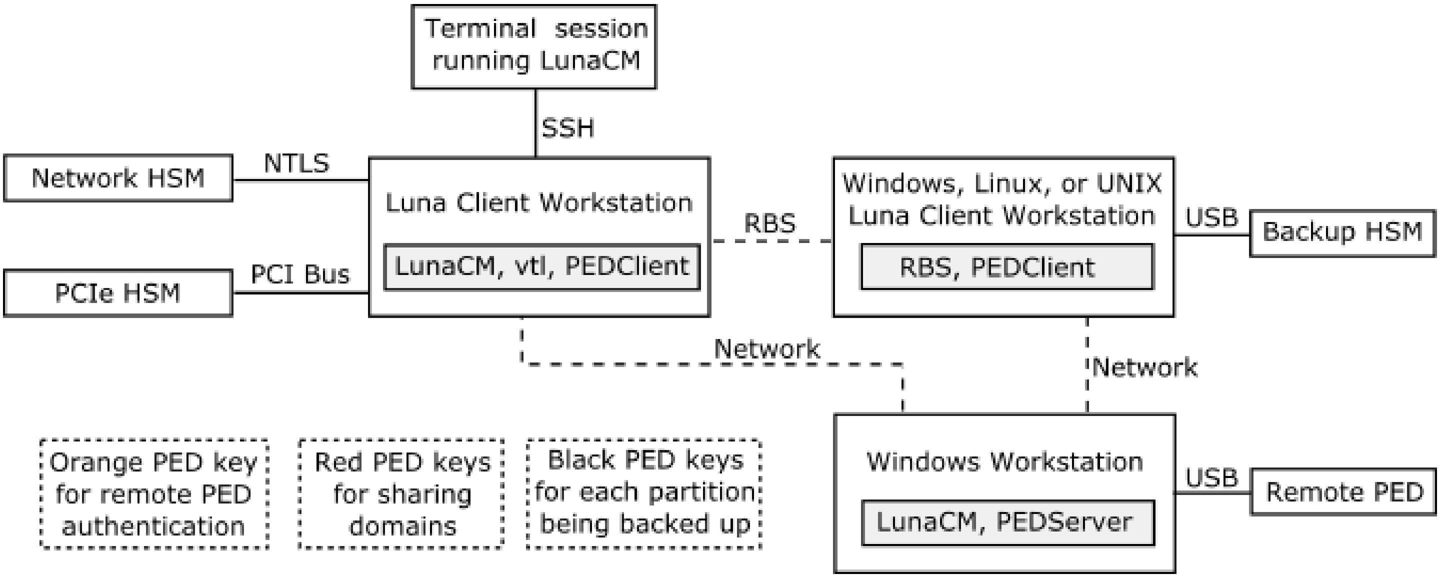

Terminal session running LunaCM
SSH
Network HSM
NTLS
PCIe HSM
PCI Bus
Luna Client Workstation
LunaCM, vtl, PEDClient
RBS
Windows Luna Client Workstation
RBS, PEDServer, PEDClient
USB
Backup HSM
USB
Remote PED
Orange PED key for remote PED authentication
Red PED keys for sharing domains
Black PED keys for each partition being backed up

Figure 5: Configuration for remote backup of a SafeNet Luna HSM client workstation slot with the remote PED connected to a separate workstation


Terminal session running LunaCM
SSH
Network HSM
NTLS
PCIe HSM
PCI Bus
Luna Client Workstation
LunaCM, vtl, PEDClient
RBS
Windows, Linux, or UNIX Luna Client Workstation
RBS, PEDClient
USB
Backup HSM
Network
Network
Windows Workstation
LunaCM, PEDServer
USB
Remote PED
Orange PED key for remote PED authentication
Red PED keys for sharing domains
Black PED keys for each partition being backed up

Configuring the Remote Backup Service (RBS)

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. 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. Running RBS also requires running PED Client on that computer, as well as on the distant primary - the paired instances of PED Client form the communications link that makes RBS possible.

RBS requires PED Client on both the RBS client and RBS server ends.

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

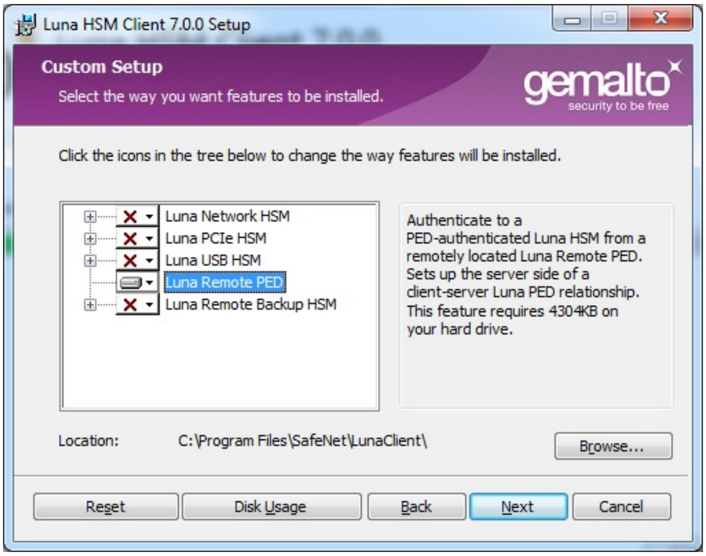
However, PEDClient is also used in the communication component of Remote Backup Service. So, PEDClient should run on all the platforms that have HSMs - where a SafeNet Luna USB HSM or SafeNet Luna PCIe HSM is installed (PEDClient is already inside SafeNet Luna Network HSM 5.2 and newer...) - and also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

To configure RBS:

1. Install the SafeNet Luna HSM client software on the computer used to manage the HSMs/partitions you want to back up. If you use PED authentication, ensure that the Remote PED option is installed. You must also install the SafeNet Luna Network HSM client software in addition to the SafeNet Luna USB HSM or SafeNet Luna PCIe HSM software, because the SafeNet Luna Network HSM client is the only one that includes the **vtl** utility, which is required to perform the certificate exchange that enables Remote Backup Service.



Luna HSM Client 7.0.0 Setup

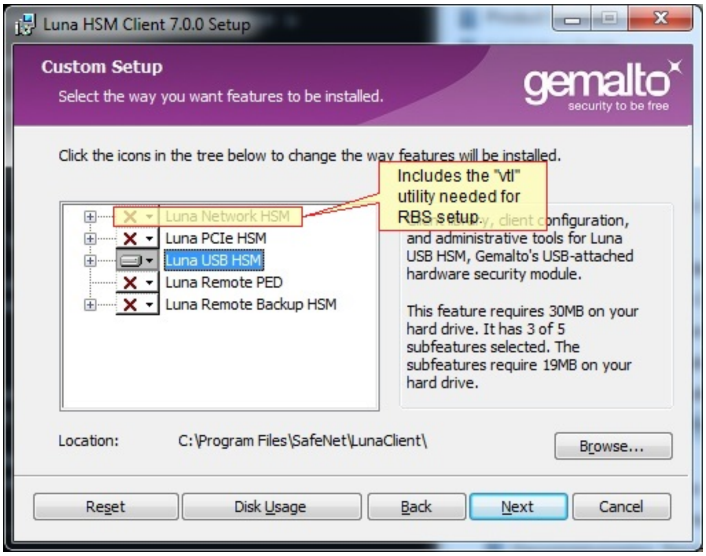
Custom Setup

Select the way you want features to be installed.

Click the icons in the tree below to change the way features will be installed.

Authenticate to a PED-authenticated Luna HSM from a remotely located Luna Remote PED. Sets up the server side of a client-server Luna PED relationship. This feature requires 4304KB on your hard drive.

Location: C:\Program Files\SafeNet\LunaClient\



Luna HSM Client 7.0.0 Setup

Custom Setup

Select the way you want features to be installed.

Click the icons in the tree below to change the way features will be installed.

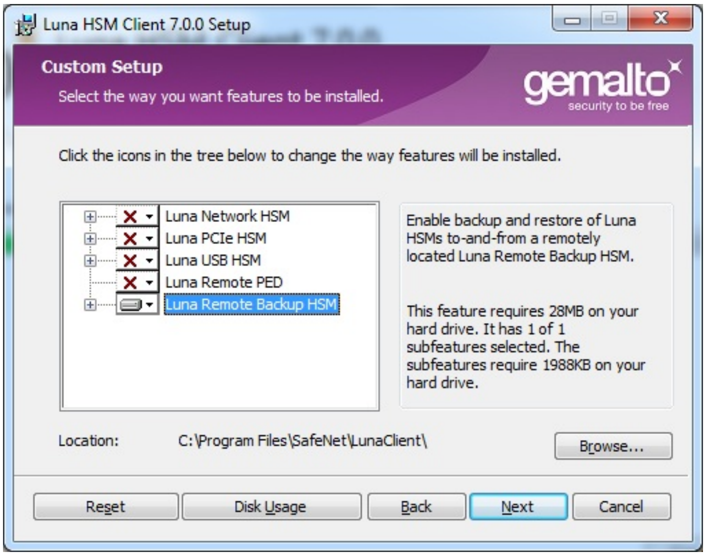
Includes the "vtl" utility needed for RBS setup.

...y, client configuration, and administrative tools for Luna USB HSM, Gemalto's USB-attached hardware security module.

This feature requires 30MB on your hard drive. It has 3 of 5 subfeatures selected. The subfeatures require 19MB on your hard drive.

Location: C:\Program Files\SafeNet\LunaClient\

2. Install the SafeNet Luna HSM client software on the workstation used to host your Backup HSM. Select the Remote Backup option. If the workstation is running Windows, and will be used to connect a Remote PED, install the Remote PED option here.



Luna HSM Client 7.0.0 Setup

Custom Setup

Select the way you want features to be installed.

Click the icons in the tree below to change the way features will be installed.

Enable backup and restore of Luna HSMs to-and-from a remotely located Luna Remote Backup HSM.

This feature requires 28MB on your hard drive. It has 1 of 1 subfeatures selected. The subfeatures require 1988KB on your hard drive.

Location: C:\Program Files\SafeNet\LunaClient\

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 the 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. Create the client certificate (if not already done) with **vtl createCert -n <host_ip_address>**.

7. Use **scp** (Unix/Linux) or **pscp** (Windows) to copy the certificate generated earlier (**server.pem**) to your primary HSM host computer (or SafeNet Luna Network 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%
```

8. 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
 Server: 192.20.9.253
```

> **NOTE** If you encounter problems, try changing the RBS and PED Client ports from the default values. Check that your firewall is not blocking ports used by the service. (Refer to the command syntax pages for default values.)

Backing Up an Application Partition to a Remotely Located Backup HSM

This section describes how to backup an application partition to a remotely located Backup HSM using RBS.

Prerequisites

You will need the following components to perform a remote backup:

Quantity	Description
1	SafeNet Luna HSM 5.2 or newer
1	Windows computer with SafeNet Luna Network 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 (Remote PED with f/w 2.7.1 or later)*
1	Power cable for Luna PED (Remote)
2	USB to mini USB cable for Luna PED (Remote) and SafeNet Luna Backup HSM

> **NOTE** The Luna PED that is connected to the Windows computer, in order to perform Remote PED operations with the distant SafeNet Luna Network HSM appliance, must be a Luna PED (remote-capable version) and is used in Remote mode and in Local mode. You also have the option to connect a second Luna 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 Luna PED connected to the workstation in Remote mode.

Assumptions

The following examples assume that you have set up RBS, as described in "Configuring the Remote Backup Service (RBS)" on page 64, and have prepared for the backup, as follows:

- The Backup HSM and the HSMs/partitions you want to back up are initialized with appropriate keys (blue SO and black Partition Owner/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 Network HSM, NTLS is established between your workstation computer, acting as a SafeNet Luna Network HSM client, and the distant SafeNet Luna Network HSM - that is, the workstation is registered as a client with the partition.
- A Remote PED session key (orange RPV key) has been created and associated with the distant SafeNet Luna HSM.

To Backup an Application Partition to a Remotely Located Backup HSM:

The following procedure provides an example illustrating how to remotely backup a PED-authenticated application partition. In this example a single remote PED, attached to the Windows workstation used to host the Backup HSM, is used.

Set up the remote PED

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)

3. At the Remote Luna PED (Luna PED with remote capability, connected to the USB port of the workstation), do the following:
   - Press **<** on the PED keypad to navigate to the main menu.
   - Press **7** to enter Remote mode.
4. Run **PedServer** to start the Remote PED service on the administrative workstation (Windows) computer, as follows:
   - In a Command Prompt (DOS) window, change directory to the location of the **PEDServer.exe** file and run that file:

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

5. Open an administrative connection (SSH) to the distant SafeNet Luna HSM (for SafeNet Luna Network 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 <workstation_ip_address> -port 1503

or
lunacm:> hsm ped connect -ip <workstation_ip_address> -port 1503
```

   Insert the orange RPV PED key that matches the RPV of the distant SafeNet Luna HSM.
   The Remote PED Client in the SafeNet Luna Network HSM appliance or in the SafeNet Luna HSM client workstation establishes a connection with the listening PedServer on your remote PED workstation.

Backup a slot to the remotely located backup HSM

> **NOTE** The following steps apply to LunaCM only. For LunaSH, follow the procedure "To backup a SafeNet Luna Network HSM partition to a directly connected Backup HSM:" on page 1. Use the **token backup list** and **token backup show** commands to ensure that the remote Backup HSM is visible.

6. 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) 2006-2017 Gemalto, Inc.

        Available HSM's:

        Slot Id ->              1
        HSM Label ->            SA82_P1
        HSM Serial Number ->    16298193222733
        HSM Model ->            LunaSA 7.0.0
        HSM Firmware Version -> 7.0.1
        HSM Configuration ->    Luna User Partition, With SO (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 Network 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 -> 6.26.01
        HSM Configuration ->    Luna HSM (PED) Backup Device
        HSM Status ->           OK

        Current Slot Id: 1
```

7. If the current slot is not the slot that you wish to backup, use the **slot set** command to go to the correct slot.

```
lunacm:> slot set slot 1

        Current Slot Id: 1     (Luna User Slot 6.22.0 (PED) Signing With Cloning Mode)

Command Result : No Error
```

8. Establish that the HSM is listening for the remote Luna PED at the correct location:

> **NOTE** 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
```

9. 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 the activated state.

```
lunacm:> role login -name Crypto Officer

        Option -password was not supplied.  It is required.

        Enter the password: *******

        User is activated, PED is not required.

Command Result : No Error
```

10. Disconnect the PED from your source HSM (slot 1 in this example), and connect to the remote 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
```

11. Perform the backup from the current slot to the partition that you designate on the Remote 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 archive 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 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
```

12. The backup operation is complete.

Restoring an HSM Partition From a Remotely Located Backup HSM

This section describes how to restore an application partition from a remotely located Backup HSM using RBS.

To restore an application partition from a remotely located backup HSM:

The following procedure provides an example of how to restore a partition from a remotely located Backup HSM. In this example, the partition is restored to a SafeNet Luna Network HSM partition that is not in the activated state. A single remote PED is used to authenticate to the remote Backup HSM and the SafeNet Luna Network HSM partition. If your primary HSM partition (the partition onto which you will restore the backed-up objects) is in the activated state, then only the Backup HSM needs PED activity for authentication during restore.

> **NOTE** The following steps apply to LunaCM only. For LunaSH, follow the procedure "To restore a SafeNet Luna Network HSM partition from a directly connected Backup HSM:" on page 1. Use the **token backup list** and **token backup show** commands to ensure that the Remote Backup HSM is visible.

1. In our test setup, we have each of several SafeNet Luna 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) 2006-2017 Gemalto, Inc.


        Available HSMs:

        Slot Id ->              0
        Label ->
        Serial Number ->        16298193222733
        Model ->                LunaSA 7.0.0
        Firmware Version ->     7.0.1
        Configuration ->        Luna User Partition With SO (PED) Signing With Cloning Mode
        Slot Description ->     Net Token Slot

        Slot Id ->              1
        Label ->
        Serial Number ->        16298193222735
        Model ->                LunaSA 7.0.0
        Firmware Version ->     7.0.1
        Configuration ->        Luna User Partition With SO (PED) Signing With Cloning Mode
        Slot Description ->     Net Token Slot

        Slot Id ->              2
        Label ->                legacypar1
        Serial Number ->        16298193222734
        Model ->                LunaSA
        Firmware Version ->     6.22.0
        Configuration ->        Luna User Partition, No SO (PED) Signing With Cloning Mode
        Slot Description ->     Net Token Slot

        Slot Id ->              3
        Label ->                SAbck1
        Serial Number ->        700101
        Model ->                G5Backup
        Firmware Version ->     6.26.0
        Configuration ->        Luna User Partition With SO (PED) Signing With Cloning Mode
        Slot Description ->     User Token Slot
        Slot Id ->              5
        Tunnel Slot Id ->       7
        Label ->
        Serial Number ->        349297122734
        Model ->                K6 Base
        Firmware Version ->     6.22.0
        Configuration ->        Luna User Partition With SO (PED) Signing With Cloning Mode
        Slot Description ->     User Token Slot

        Slot Id ->              6
        Tunnel Slot Id ->       7
        Label ->                mypcie6
        Serial Number ->        150022
        Model ->                K6 Base
        Firmware Version ->     6.22.0
        Configuration ->        Luna HSM Admin Partition (PED) Signing With Cloning Mode
        Slot Description ->     Admin Token Slot
        HSM Configuration ->    Luna HSM Admin Partition (PED)
        HSM Status ->           OK

        Slot Id ->              8
        HSM Label ->            myG5pw
        HSM Serial Number ->    7001312
        HSM Model ->            G5Base
        HSM Firmware Version -> 6.10.4
        HSM Configuration ->    SafeNet Luna USB HSM (PW) Signing With Cloning Mode
        HSM Status ->           OK

        Current Slot Id: 0
```

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

```
lunacm:>ped get

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

Command Result : No Error
```

3. Connect to Remote PED with **ped connect**.
4. Log into the partition to which you want to restore.

> **NOTE** 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:> role login -n Crypto Officer

        enter password: *******

        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
```

(The current selected slot in LunaCM is still slot 0, and having ensured login status on that slot/partition we have just released the Remote PED connection there. The other end of the Remote PED pair, the PED-connected host computer running PedServer, is now free to accept a Remote PED link from another PedClient, which will be the host attached to the SafeNet Luna Backup HSM.)

> **NOTE** In this example, the SafeNet Luna Network HSM partition, to which we will restore objects, is visible in LunaCM at slot 0 because it is linked to this SafeNet Luna HSM client by NTLS, while this Client is registered to that partition at the SafeNet Luna Network HSM.
>
> The SafeNet Luna Backup HSM is visible in LunaCM, at slot 3 in this case, because it is linked by the RBS connection that you previously established (see "To Configure RBS" above in this chapter); that is, PedClient is running on this Client, and PedClient and rbs.exe are running on the Backup HSM's host, with each other identified as their partner in the RBS link.

5. 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 0 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
```

The **ped connect** command specifies the slot (now the SafeNet Luna Backup HSM) that makes a new Remote PED connection, because that slot indication is part of the command - and **ped get** verifies the new Remote PED-connected slot. But the focus of the library/LunaCM has not changed from slot 0; any other LunaCM commands that act on a slot will act on slot 0 until you change that with **slot set**. You could verify that current focus, if you wished, by running **slot list** again.

6. Restore to the current slot from the slot that corresponds to the Backup HSM.

```
lunacm:> partition archive 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 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
```

Because the LunaCM focus rests with the target partition in slot 0, your **partition archive restore** command must explicitly identify the slot from which backup source objects are to be cloned, slot 3 in this example, onto the target partition, current-slot 0 in this case. You also specified the backup partition name, because a SafeNet Luna Backup HSM can contain more than one archived partition.

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

```
lunacm:> partition archive 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

lunacm:>
```

8. Remote restore from backup, using RBS, is complete.

To restore onto a different remote SafeNet Luna HSM, the same arrangement is required:

- The remote HSM must already have a suitable partition.
- If the restore-target HSM is a SafeNet Luna Network HSM, the target partition can have any name - it does not need to match the name of the source partition on the backup device.
- Your workstation must be registered as a client to that partition.

Backup and Restore From the Appliance to a Local Backup HSM (LunaSH)

This section describes how to use LunaSH to backup and restore a partition on the appliance to a locally connected SafeNet Luna Backup HSM (Backup HSM). To perform a local backup, you connect the SafeNet Luna Backup HSM to a USB port on the SafeNet Luna Network HSM appliance and use LunaSH to log in as the Crypto Officer (CO) to the HSM partitions that you want to backup.

The backup operation can go from a source partition (on a SafeNet Luna Network HSM) to an existing partition on the Backup HSM, or if one does not exist, a new partition can be created during the backup. The restore operation, however, cannot create a target partition on a SafeNet Luna Network HSM; it must already exist.

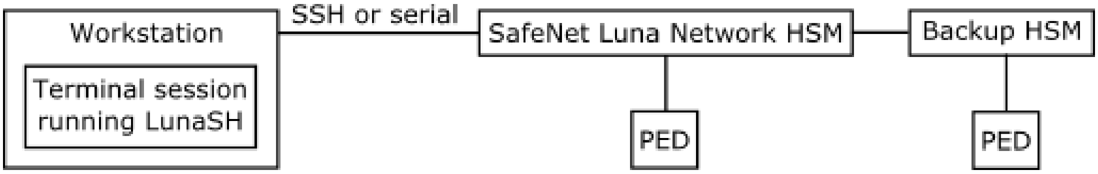
You can restore a partition backup to the source HSM or to a different SafeNet Luna Network HSM. The HSM you restore to must already have a suitable partition created for the restored objects. The partition can have any name - it does not need to match the name of the source partition on the backup HSM.

You can connect the Backup HSM directly to the SafeNet Luna Network HSM appliance to backup some or all of the individual partitions it contains, using LunaSH. You require the Partition Crypto Officer (CO) credentials for each partition you want to backup.

> **NOTE** You cannot use this method to backup partitions configured to use STC (see "Secure Trusted Channel (STC)" on page 279). To backup a partition configured to use STC, you must use LunaCM, as described in "Backup and Restore From the Appliance to a Local Backup HSM (LunaSH)" above.

To perform a backup/restore, you open an SSH or serial connection from your workstation to the appliance, and use LunaSH to perform a backup to the Backup HSM connected to the appliance, as illustrated in the following figure:

Figure 6: Partition backup/restore using a Backup HSM connected directly to the appliance



Workstation requirements

The workstation is simply a display terminal for LunaSH running on the appliance. It requires an SSH client (ssh on Linux, PuTTY on Windows). It does not require the SafeNet Luna HSM client software.

PED-authenticated partitions

The PEDs are required only if the SafeNet Luna Network HSM is PED-authenticated. The appropriate SO (blue), partition (black) and domain (red) PED keys are required. The Backup HSM and SafeNet Luna Network HSM must share the same domain (red) PED key value.

Although two PEDs are recommended (one connected to the SafeNet Luna Network HSM and one connected to the Backup HSM) you can use a single PED, if desired. If using a single PED, note that you can connect the PED to only one HSM at a time. You will need to disconnect it from the source (SafeNet Luna Network HSM) HSM and connect to the target (SafeNet Luna Backup HSM) when PED operations are needed at those HSMs respectively.

Backing Up a Partition to a Locally Connected Backup HSM

You can backup any partitions you can log in to as the Crypto Officer.

To backup a SafeNet Luna Network HSM partition to a directly connected Backup HSM:

1. Connect all the required components and open a terminal session to the SafeNet Luna Network HSM appliance. See the following topics for details:

   - "Open a Connection" on page 1 in the *Configuration Guide*
   - "Backup HSM Installation, Storage, and Maintenance" on page 51

   Connect your PED directly to the HSM, and set it to Local PED-USB mode. (For legacy PED-HSM connections via MDSM cable, set your PED to Local PED-SCP mode.) See "Changing Modes" on page 202 for instructions on changing modes on the Luna PED.

   Connect your Backup HSM to any USB port on the appliance.

2. Open a LunaSH session on the SafeNet Luna Network HSM appliance.

```
login as: admin
admin@192.20.10.202's password:
Last login: Tue Dec 30 16:03:46 2014 from 192.16.153.111

SafeNet Luna Network HSM 7.0 Command Line Shell - Copyright (c) 2001-2017 Gemalto, Inc. All
rights reserved.
[myluna] lunash:>
```

3. Use the **token backup list** and **token backup show** commands to determine the serial number of the Backup HSM and to verify its partition and storage configuration:

```
lunash:>token backup list

Token Details:
============
Token Label:                      BackupHSM
Slot:                             6
Serial #:                         7000179
Firmware:                         6.26.0
Hardware Model:                   G5 Backup

Command Result : 0 (Success)


lunash:> token backup show -serial 700179
```

```
Token Details:
============
Token Label:                        BackupHSM
Serial #:                           700179
Firmware:                           6.22.0
Hardware Model:                     SafeNet Luna USB HSM
Authentication Method:              PED keys
Token Admin login status:           Logged In
Token Admin login attempts left:    3 before Token zeroization!

Partition Information:
======================
Partitions licensed on token:       20
Partitions created on token:        0
----------------------

There are no partitions.

Token Storage Information:
==========================

Maximum Token Storage Space (Bytes): 16252928
Space In Use (Bytes):                0
Free Space Left (Bytes):             16252928

License Information:
====================

621010355-000      621-010355-000 G5 Backup Device Base
621000005-001      621-000005-001 Backup Device Partitions 20
621000006-001      621-000006-001 Backup Device Storage 15.5 MB
621000007-001      621-000007-001 Backup Device Store MTK Split Externally
621000008-001      621-000008-001 Backup Device Remote Ped Enable

Command result : 0 (Success)
```

4. Use the **partition backup** command to backup a specified partition and provide the PED keys as prompted, for example:

```
[myluna] lunash:>par backup -s 7000179 -par p1 -tokenPar bck1

Type 'proceed' to continue the backup, or 'quit'
           to abort this operation.
> proceed
Please enter the password for the HSM partition:
> *******

Warning: You will need to attach Luna PED to the SafeNet Luna Backup HSM
         to complete this operation.
         You may use the same Luna PED that you used for SafeNet Luna Network HSM.

Please hit <enter> when you are ready to proceed.

Luna PED operation required to login to token - use token Security Officer (blue) PED key.
Luna PED operation required to create a partition - use User or Partition Owner (black) PED
key.
Luna PED operation required to login to user on token - use User or Partition Owner (black) PED
```

```
key.
Luna PED operation required to generate cloning domain on the partition - use Domain (red) PED
key.

Object "1-User DES Key1" (handle 17) cloned to handle 11 on target
Object "1-User DES Key2" (handle 18) cloned to handle 12 on target
Object "1-User Public RSA Key1-512" (handle 19) cloned to handle 13 on target
.
.
.
Object "1-User ARIA Key3" (handle 124) cloned to handle 118 on target
Object "1-User ARIA Key4" (handle 125) cloned to handle 119 on target
Object "1-User ARIA Key5" (handle 126) cloned to handle 120 on target
'partition backup' successful.

Command Result : 0 (Success)
```

5. Use the **token backup show** command to verify the backup:

```
lunash:> token backup show -serial 667788
Token Details:
============
Token Label:                      BackupHSM
Serial #:                         700179
Firmware:                         6.26.0
HSM      Model:                   G5Backup
Authentication Method:            PED keys
Token Admin login status:         Logged In
Token Admin login attempts left:  3 before Token zeroization!

Partition Information:
======================
Partitions licensed on token:     20
Partitions created on token:      1
----------------------
Partition: 7000179008,            Name: bck1.

Token Storage Information:
==========================

Maximum Token Storage Space (Bytes):   16252928
Space In Use (Bytes):                     43616
Free Space Left (Bytes):               16209312

License Information:
====================

621010355-000      621-010355-000 G5 Backup Device Base
621000005-001      621-000005-001 Backup Device Partitions 20
621000006-001      621-000006-001 Backup Device Storage 15.5 MB
621000007-001      621-000007-001 Backup Device Store MTK Split Externally
621000008-001      621-000008-001 Backup Device Remote PED Enable

Command result : 0 (Success)
```

Restoring a Partition from a Locally Connected Backup HSM

You can backup any partitions you can log in to as the Crypto Officer.

To restore a SafeNet Luna Network HSM partition from a directly connected Backup HSM:

To restore the partition contents from the SafeNet Remote Backup Device to the same local SafeNet Luna Network HSM, use the same setup described above, but use the **partition backup restore** command instead.

1. Connect all the required components and open a terminal session to the SafeNet Luna Network HSM appliance. See the following topics for details:
   - "Open a Connection" on page 1 in the *Installation and Configuration Guide*
   - "Backup HSM Installation, Storage, and Maintenance" on page 51

   Connect your PED directly to the HSM, and set it to Local PED-USB mode. (For legacy PED-HSM connections via MDSM cable, set your PED to Local PED-SCP mode.) See "Changing Modes" on page 202 for instructions on changing modes on the Luna PED.
2. Open a LunaSH session on the SafeNet Luna Network HSM appliance.

```
login as: admin
admin@192.20.10.202's password:
Last login: Tue Feb 28 16:03:46 2012 from 192.16.153.111

SafeNet Luna Network HSM 7.0 Command Line Shell - Copyright (c) 2001-2016 Gemalto, Inc. All
rights reserved.
[myluna] lunash:>
```

3. Use the **partition restore** command to restore a partition:

```
[myluna] lunash:>par restore -s 7000179 -tokenPar bk5 -par p1 -replace
Please enter the password for the HSM partition:
> *******

  CAUTION:  Are you sure you wish to erase all objects in the
          partition named:              p1
          Type 'proceed' to continue, or 'quit' to quit now.
          > proceed
Warning:  You will need to attach Luna PED to the SafeNet Luna Backup HSM to complete this
operation.
                 You may use the same Luna PED that you used for SafeNet Luna Network HSM.


Please hit <enter> when you are ready to proceed.

Luna PED operation required to login to user on token - use User or Partition Owner (black) PED
key.
Object "1-User DES Key1" (handle 17) cloned to handle 11 on target
Object "1-User DES Key2" (handle 18) cloned to handle 12 on target
Object "1-User Public RSA Key1-512" (handle 19) cloned to handle 13 on target
.
.
.
Object "1-User ARIA Key3" (handle 124) cloned to handle 118 on target
Object "1-User ARIA Key4" (handle 125) cloned to handle 119 on target
Object "1-User ARIA Key5" (handle 126) cloned to handle 120 on target
```

```
'partition restore' successful.

Command Result : 0 (Success)
```

Troubleshooting

This section provides troubleshooting tips for errors you may encounter when performing a partition backup/restore operation.

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

If you insert a backup token that has previously been used on a password-authenticated SafeNet Luna Network HSM into a PED-authenticated SafeNet Luna Network HSM, and attempt to initialize it, the system responds with the message "Warning: This token is not in the factory reset (zeroized) state" as shown in the following example:

```
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)
```

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 the **token factoryreset** command, and then initialize the token, as shown in the following example:

```
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)

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)
```

CHAPTER 3: Capabilities and Policies

The SafeNet Luna Network HSM's configuration is based on HSM capabilities, displayed using the LunaSH command **hsm showpolicies**. 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 86

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 Network 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, login 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.

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 Network HSMs.
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.

#	HSM Capability	HSM Policy	Description

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 show** in LunaSH:
			`FIPS 140-2 Operation:`
			`=====================`
			`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 Logins" on page 363 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 Network HSM can use the Korean algorithm set. This capability may be purchased as an upgrade. See "HSM Capability and Partition Upgrades" on page 310.
18	FIPS evaluated		Always disallowed - deprecated policy. All SafeNet Luna Network HSMs are capable of operating in FIPS Mode.
19	Manufacturing Token		N/A (SafeNet internal use only)
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.

#	HSM Capability	HSM Policy	Description
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 Network 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 Network 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 Network HSM you selected at time of purchase. On some models, the number of allowable partitions can be upgraded with a separate purchase. See "HSM Capability and Partition Upgrades" on page 310 for more information.
35	Enable Single Domain		Not applicable to SafeNet Luna Network HSMs.
36	Enable Unified PED Key		Not applicable to SafeNet Luna Network HSMs.
37	Enable MofN	Allow MofN	If allowed on PED-authenticated SafeNet Luna Network HSMs, this policy enables you to split a PED secret among multiple PED keys (see "M of N Split Secrets" on page 199). 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.

#	HSM Capability	HSM Policy	Description
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 279). If disallowed, all partition-client connections must use NTLS.
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 326 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.
45	Enable Fast-Path		Not available in this release.
46	Allow Disabling Decommission	Disable Decommission	If enabled, the decommission button is disabled, preventing decommissioning of the HSM. **Note:** You cannot enable this policy if HSM policy 40: Decommission on Tamper is enabled. **CAUTION:** Changing this policy will destroy partitions on the HSM, and they must be recreated. **Destructive: OFF-to-ON, ON-to-OFF**
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 326 for more information.

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**. 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 off. Partition policies **0** and **1** may not be set to **1** (ON) at the same time. Default: **OFF** **Destructive: OFF-to-ON**

#	Partition Capability	Partition Policy	Description
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 Network 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**
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 Network HSMs. The Partition SO cannot change this policy. Default: always **OFF**

#	Partition Capability	Partition Policy	Description
10	Enable multipurpose keys	Allow multipurpose keys	If enabled, keys for multiple purposes, such as signing and decrypting, may be created on the partition. The Partition SO can turn this feature on or off. If disabled, keys created on (or unwrapped onto) the partition must specify only a single function in the attribute template. 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**
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 185 and "Failed Logins" on page 363 for more information. Default: **ON** **Destructive: OFF-to-ON**

#	Partition Capability	Partition Policy	Description
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**
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 Logins" on page 363). 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**

#	Partition Capability	Partition Policy	Description
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 326, and "Activation and Auto-Activation on PED-Authenticated Partitions" on page 185 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 326, and "Activation and Auto-Activation on PED-Authenticated Partitions" on page 185 for more information. Default: **OFF**
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**

#	Partition Capability	Partition Policy	Description
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**
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**

#	Partition Capability	Partition Policy	Description
37	Enable Secure Trusted Channel	Force Secure Trusted Channel	If enabled, the Partition SO can turn this policy on to require Secure Trusted Channel (STC) for partition access. If disabled, the Client will use NTLS to access the partition. **NOTE:** It is not possible for a single Client to access some partitions on an appliance using STC and others on the same appliance using NTLS. All connections between a single client and a single SafeNet Luna Network HSM must be either STC or NTLS. See "Secure Trusted Channel (STC)" on page 279 in the *Administration Guide* for more information. 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.

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 changepolicy** in LunaSH 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 LunaSH.

To zeroize the HSM and keep the current policy settings, use **hsm zeroize** in LunaSH.

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" on the next page
- "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. Login to LunaSH as **admin**. If you are creating a template from an initialized HSM, you must log in as HSM SO.

   lunash:>**hsm login**

2. Create the HSM policy template file with an original filename. No file extension is required. If a template file with the same name exists, it is overwritten.

   lunash:>**hsm showpolicies -exporttemplate** <filename>

   ```
   lunash:>hsm showpolicies -exporttemplate HSMPT


   HSM policies successfully written.

   Use 'scp' from a client machine to get file named:
   HSMPT


   Command Result : 0 (Success)
   ```

3. On a client workstation, use **scp**/**pscp** to transfer the template file from the source appliance.
4. 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 86 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, the **-exporttemplate** option is not available.
- If you are using Secure Trusted Channel (STC) client connections, you cannot use partition policy templates.
- The following restrictions apply when configuring partitions for Cloning or Key Export (see "Keys In Hardware vs. Private Key Export" on page 99 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. From a client workstation, use **scp**/**pscp** to transfer the template file to the destination appliance.
2. Login to LunaSH as **admin** on the destination appliance, and initialize the HSM using the policy template file.

   lunash:>**hsm init -label** <label> **-applytemplate** <filename>

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

   lunash:>**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: Keys In Hardware vs. Private Key Export

By default, the SafeNet Luna Network 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.

The Partition SO can set the mode by changing the following policies (see "Partition Capabilities and Policies" on page 86 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 101

> **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 5: 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 Chrystoki.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.

> **NOTE** For SafeNet Luna Network HSM, LunaSH commands use onboard default configuration settings. Clients that are sent to the HSM via SafeNet Luna HSM Client, making use of the client library, include the relevant configuration settings from the client-side Chrystoki.conf /Crystoki.ini configuration file.

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.

Setting	Range (Default)	Description
LibNT32=	(<luna_client_ dir>\win32\libCryptoki2.dll)	Path to the Chrystoki2 library on 32-bit Windows systems only.
[Luna]		
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, CommandTimeOutPedSet = 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** 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. On the SafeNet Luna Network HSM appliance,

Setting	Range (Default)	Description
		where you do not have direct access to the file system, the TCPKeepAlive= setting is controlled by the LunaSH command **ntls tcp_ keepalive set**. The settings at the appliance and the client are independent. This allows a level of assurance, in case (for 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**).

Setting	Range (Default)	Description

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]		
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
LegacyMemoryReportingMode=	(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.

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.
CopyRSAPublicValuesFromPrivateTemplat e	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
[Secure Trusted Channel]		
ClientIdentitiesDir=	<luna_client_dir>\data\client_identities	Specifies the directory used to store the STC client identity
PartitionIdentitiesDir=	<luna_client_dir>\data\partition_identities	Specifies the directory used to store the STC partition identities exported using the LunaCM **stcconfig partitionidexport** command
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.
SoftTokenDir=	<luna_client_dir>**\softtoken**	Specifies the location where the STC client soft token (**token.db**) is stored. Each client soft token is stored in its own numbered subdirectory. **Note:** In this release there is only one client token, which is stored in the **001** subdirectory.

CHAPTER 6: Decommissioning, Zeroizing, 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:

Decommissioning the HSM Appliance

This section describes how to decommission the appliance to remove all current key material and configurations, so that it can be safely redeployed.

To decommission a SafeNet Luna Network HSM:

For full decommission (removing the unit from service, clearing the HSM of all your material, clearing the appliance of all identifying information) of a SafeNet Luna Network HSM appliance, and assuming that you can power the appliance and gain admin access, follow these steps in LunaSH, using a serial connection:

1. Rotate all logs:

   lunash:> **syslog rotate**

2. Delete all files in the SCP directory:

   lunash:> **my file clear**

3. Delete all logs:

   lunash:> **syslog cleanup**

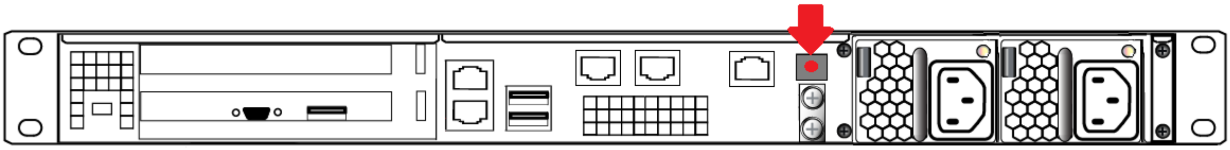
4. Return the appliance to factory-default settings:

   lunash:> **sysconf config factoryreset -service all**

5. Delete any backups of settings:

   lunash:> **sysconf config clear**

6. Push the decommission button (small red button, inset in the SafeNet Luna Network HSM back panel).



7. Power down the appliance.

8. Power up the appliance. At this point, the HSM internally issues and executes a **zeroize** command to erase all partitions and objects. This step takes about five minutes. The KEK is already gone at that point – erased as soon as the button is pressed – so the step of erasing partitions and objects is for customers subject to especially rigid decommission protocols.

Disabling Decommissioning

You can disable the decommissioning feature if you have the factory-installed **HSM Capability 46: Allow 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 326). If decommissioning is disabled, the SafeNet Luna Network HSM has an indefinite shelf life, as far as the battery is concerned.

To disable decommissioning:

1. Ensure that the Disable Decommissioning capability update (CUF) is installed on the HSM. To verify that the CUF is installed, enter the following command:

   lunash:> **hsm showpolicies**

   If the CUF is installed, **HSM Capability 46: Allow Disable Decommission** and **HSM Policy 46: Disable Decommission** are listed. If they are not, contact technical support to obtain the Disable Decommissioning capability update (CUF).

2. Enter the following command to enable **HSM Policy 46: Disable Decommission**

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

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 > Run the LunaSH command **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	Run the LunaSH command **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 HSM as though it were new from the factory.

Resetting to Factory Condition

These instructions will allow you to restore your SafeNet Luna Network 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 and appliance. Reverting of software and firmware is outside their scope. You must access LunaSH via a serial console to execute **hsm factoryreset**.

To reset the HSM to factory condition:

1. Login as HSM SO.

   hsm login

2. Reset the HSM to factory settings.

   hsm factoryreset

3. Reset the appliance configuration (network settings, ssh, ntls, etc.) to factory settings.

   sysconf config factoryreset -service all

4. Reboot the appliance.

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.

Our customers are often very high-security establishments that have rigorous protocols for removing a device from service. In such circumstances, it is not sufficient to merely ensure that all material is gone from the HSM. It is also necessary to clear any possible evidence from the appliance that contains the HSM, such as IP configuration and addresses, log files, etc.

If you have any concern that simply pressing the Decommission button and running **sysconf config factoryreset** is not sufficient destruction of potentially-sensitive information, then please refer to "Decommissioning the HSM Appliance" on page 120.

Comparison of Destruction/Denial Actions

Various operations on the SafeNet Luna 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)

Scenario 3: Appliance admin password reset

Event	Scen. 1	Scen. 2	Scen. 3	How to discover (See Note 3)	How to recover
> Three bad SO login attempts > lunash:> **hsm zeroize** > lunash:> hsm **factoryreset** > Any change to a destructive policy > Firmware rollback (See Note 4)	NO	YES	NO	> Syslog entry > "HSM IS ZEROIZED" in HSM Details (from **hsm show** command)	Restore HSM objects from Backup

Event	**Scen. 1**	**Scen. 2**	**Scen. 3**	**How to discover** (See Note 3)	**How to recover**
Login to SafeNet Luna Network HSM "recover" account (local serial connection)	NO	NO	YES	Syslog entry shows login by "recover"	Log into appliance as admin, using the reset password "PASSWORD" and change to a secure password
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 > lunash:> **hsm stm transport**	YES	NO	NO	Parse Syslog 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 Syslog like: "HSM internal error", "device error" SafeNet Luna Network HSM appliance front panel flashes error 30.	Reboot [See Note 1]
Decommission > Pressing the Decommission button on the back of the appliance	NO	YES	NO	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 165.

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 Thalesrepresentative 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 Network HSM before returning it to Thales Group:

- Decommission the HSM, forcibly clearing all HSM contents (see "Decommissioning the HSM Appliance" on page 120 for instructions)
- Set the HSM into Secure Transport Mode (see "Secure Transport Mode" on page 275 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.

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
- press the Decommission button on the SafeNet Luna Network HSM back panel
- 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

CHAPTER 7: High-Availability (HA) Configuration and Operation

This chapter describes how to configure and use SafeNet Luna HSMs to provide load-balancing and redundancy for mission-critical applications. It contains the following sections:

High Availability (HA) Overview

You can use the SafeNet Luna HSM client to group multiple devices, or partitions, into a single logical group – known as an HA (High Availability) group. When you create an HA group, it is listed as a virtual HA slot in the client. Any applications that use the virtual HA slot can access cryptographic services as long as at least one member of the HA group remains functional and connected to the application server. In addition, the client performs load balancing among the HA group members, allowing many cryptographic commands to be automatically distributed across the HA group, and enabling linear performance gains for many applications.

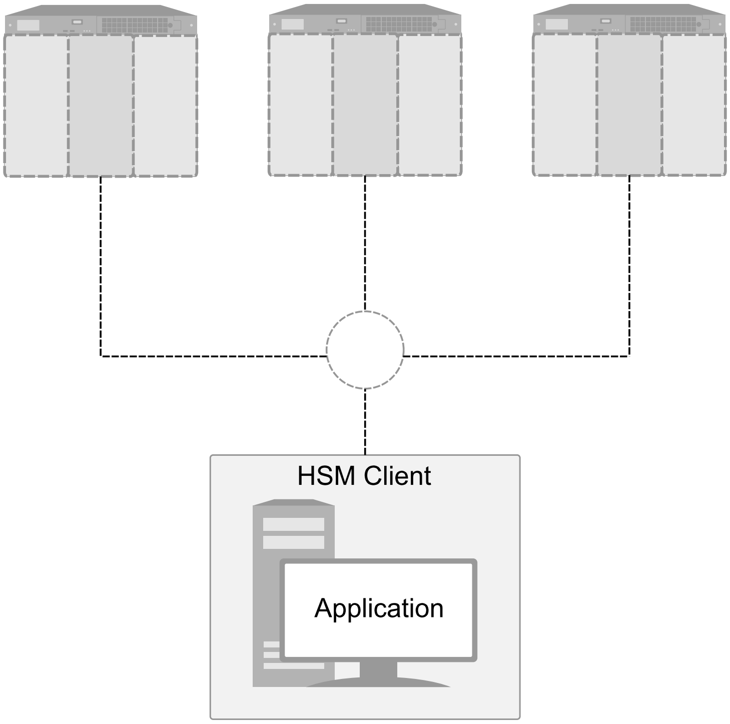
How HA is Implemented

The HA and load-balancing functionality is implemented in the SafeNet Luna HSM client, and uses the cloning function to replicate/synchronize content across HA-group members. There is no direct connection between the members of an HA group, and all communications between the members of an HA group are managed by the client. The HSMs and appliances are not involved and, except for being instructed to clone objects to certain HSMs during a synchronization operation, are unaware that they might be configured in an HA group. The advantage of this approach is that it allows you to configure HA groups on a per-application (or per-slot) basis.

To create an HA group, you must first register your client with each HSM you want to include in the HA group. You then use the client-side administration commands to define the HA group and set any desired configuration options. You can configure several options including:

- Setting automatic or manual recovery mode
- Setting some HSMs as standby members
- Performing various manual synchronization and recovery operations

Once defined, the SafeNet Luna HSM client presents the HA group as a virtual slot, which is a consolidation of all the physical HSMs in the HA group. Any operations that access the slot are automatically distributed between the group members, to provide load balancing, and all key material is automatically replicated and synchronized between each member of the HA group.



Example: Database Encryption

This section walks through a specific sample use case of some of the HA logic with a specific application – namely 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. Refer to the backup section of this manual for proper backup procedures.

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.

With this in mind, before every re-key event the user should ensure the HA group has sufficient redundancy. A re-key will succeed so 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, SafeNet 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. As such, requests to encrypt or decrypt table keys are distributed across the entire group. So 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. With that said, 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 the primary HSM and then, even though they exist on the HSM for only a moment, they are replicated to the entire HSM group as part of the availability logic. These events are infrequent enough that this extra replication has minimal impact.

Conclusion

The SafeNet high availability and load balancing features provide an excellent set of tools to scale applications and manage availability of cryptographic services without compromising the integrity of cryptographic keys. A broad range of deployment options are supported that allow solution architects to achieve the availability needed in a manner that optimizes the cost and performance without compromising the assurance of the solution.

Load Balancing

The default behavior of the client library is to attempt to load-balance the application's cryptographic requests across each active member of an HA group. Any standby members in the HA group are not used to perform cryptographic operations, and are therefore not part of the load-balancing scheme (see "Standby Members" on page 142).

The top-level algorithm is a round-robin scheme that is modified to favor the least busy device in the set. As each new command is processed, the SafeNet Luna HSM client looks at how many commands it has scheduled on every device in the group. If all devices have an equal number of outstanding commands, the new command is scheduled on the next device in the list – creating a round-robin behavior. However, if the devices have a different number of commands outstanding on them, the new command is scheduled on the device with the fewest commands queued – creating a least-busy behavior. This modified round-robin has the advantage of biasing load away from any device currently performing a lengthy command. In addition to this least-busy bias, the type of command also affects the scheduling algorithm, as follows:

- Single-part (stateless) cryptographic operations are load-balanced.
- Multi-part (stateful) commands that involve cryptographic operations are load-balanced.
- Multi-part (stateful) commands that involve information retrieval are not load-balanced. Multi-part operations carry over several individual commands. The cost of distributing the commands to different HA group members is generally greater than the benefit. For this reason, multi-part commands that involve information retrieval are all targeted at one member. Multi-part operations are typically not used, or are infrequent actions, so most applications are not affected by this restriction.
- Key management commands are not load-balanced. Key management commands affect the state of the keys stored in the HSM. As such, these commands are targeted at all HSMs in the group. That is, the command is performed on the primary HSM and then the result is replicated to all members in the HA group. Key management operations are also an infrequent occurrence for most applications .

It is important to understand that the least-busy algorithm uses the number of commands outstanding on each device as the indication of its busyness. When an application performs a repeated command set, this method works very well. When the pattern is interrupted, however, the type of command can have an impact. For example, when the HSM is performing signing and an atypical asymmetric key generation request is issued, some number of the application’s signing commands are scheduled on the same device (behind the key generation). Commands queued behind the key generation therefore have a large latency driven by the key generation. However, the least-busy characteristic automatically schedules more commands to other devices in the HA group, minimizing the impact of the key generation.

It is also important to note that the load-balancing algorithm operates independently in each application process. Multiple processes on the same client or on different clients do not share their “busyness” information while making their scheduling choice. In most cases this is reasonable, but some mixed use cases might cause certain applications to hog the HSMs.

Finally, when an HA group is shared across many servers, different initial members can be selected while the HA group is being defined on each server. The member first assigned to each group becomes the primary. This approach optimizes an HA group to distribute the key management and/or multi-part cryptographic operation load more equally.

In summary, the load-balancing scheme used by SafeNet is a combination of round-robin and least-busy for most operations. However, as required, the algorithm adapts to various conditions and use cases so it might not always emulate a round-robin approach.

Example

When the client makes a request on a virtual HA slot, the request goes to the first member in the HA group, as listed in the **Chrystoki.conf** file (Linux/UNIX) or **Crystoki.ini** file (Windows), unless it is busy. A member is busy if it has not yet responded to the most recent request that was sent to it. If the primary member is busy, the client sends the request to the next non-busy member of the HA Group.

If you add network latency, or if you increase the key-size, or if you interleave other crypto operations, then performance may drop for individual active members as they become busier.

If you have any group members set to "Standby" status, then they do not contribute to group performance, even if the client can saturate the active members.

Key Replication

Whenever an application creates key material, the HA functionality transparently replicates the key material to all members of the HA group before reporting back to the application that the new key is ready. The HA library always starts with what it considers its primary HSM (initially the first member defined in an HA group). Once the key is created on the primary it is automatically replicated to each member in the group. If a member fails during this process the key replication to the failed member is aborted after the fail-over time out. If any member is unavailable during the replication process (that is, the unit failed before or during the operation), the HA library keeps track of this and automatically replicates the key when that member rejoins the group . Once the key is replicated on all active members of the HA group a success code is returned to the application.

Whether automatic or manual, object replication security is based on the use of the SafeNet cloning protocol to provide mutual authentication, confidentiality and integrity for each object that is copied from one partition to another. When partition objects are synchronized, the SafeNet Luna HSM client is used as a secure conduit to coordinate the duplication of these objects across all partitions. An object created on LunaA partition#1A is duplicated on LunaB Partition#1B using the following process:

1. The object is created on LunaA.
2. The duplicated object is then encrypted using a key derived from common Domain material (Red key) shared by each SafeNet Luna HSM in the HA group.
3. LunaA transfers the encrypted object to the SafeNet Luna Client utilizing the encrypted NTL connection between itself and the client (the object is now double encrypted).
4. The client then securely transfers the object to LunaB.
5. LunaB decrypts the object and stores it in the partition

The cloning protocol is such that it must be invoked separately for each object to be cloned and the sequence of calls required to implement the protocol must be issued by an authorized client library (residing on a client platform that has been authenticated to each of the SafeNet Luna HSMs involved in the HA group). This ensures that the use of the 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.

Manual Synchronization

To manually synchronize the contents of the members of an HA group, use the LunaCM command **hagroup synchronize**.

Failover

When an HA group is running normally the client library continues to schedule commands across all members as described above. The client continuously monitors the health of each member at two different levels:

- First, the connectivity with the member is monitored at the networking layer. Disruption of the network connection invokes a fail-over event within a twenty second timeout.
- Second, every command sent to a device is continuously monitored for completion. Any command that fails to complete within twenty seconds also invokes a fail-over event. Most commands are completed within milliseconds. However, some commands can take extended periods to complete – either because the command itself is time-consuming (for example, key generation), or because the device is under extreme load. To cover these events the HSM automatically sends “heartbeats” every two seconds for all commands that have not completed within the first two seconds. The twenty second timer is extended every time one of these heartbeats arrives at the client, thus preventing false fail-over events.

A failover event involves dropping a device from the available members in the HA group. All commands that were pending on the failed device are transparently rescheduled on the remaining members of the group. When a failure occurs, the application experiences a latency stall on some of the commands in process (on the failing unit) but otherwise sees no impact on the transaction flow. Note that the least-busy scheduling algorithm automatically minimizes the number of commands that stall on a failing unit during the twenty second timeout.

If the primary unit fails, clients automatically select the next member in the group as the new primary. Any key management or single-part cryptographic operations are transparently restarted on a new group member. In the event that the primary unit fails, any in-progress, multi-part, cryptographic operations must be restarted by the application, as the operation returns an error code.

As long as one HA group member remains functional, cryptographic service is maintained to an application no matter how many other group members fail. As discussed in "Failover" on the previous page, members can also be put back into service without restarting the application.

How Do You (or Software) Know That a Member Has Failed?

When an HA Group member first fails, the HA status for the group shows "device error" for the failed member. All subsequent calls return "token not present", until the member (HSM Partition or PKI token) is returned to service.

At the library level, what happens when a device fails or doesn’t respond?

The client library drops the member and continues with others. It will try to reconnect that member at a minimum retry rate of once per minute (configurable) for the number of times specified in the configuration file, and then stop trying that member. You can specify a number of retries from 3 to an unlimited number.

What happens to an application if a device fails mid-operation? What if it’s a multi-part operation?

Multi part operations do not fail over. The entire operation returns a failure (CKR_DEVICE_ERROR). Your application deals with the failure in whatever way it is coded to do so.

Any operation that fails mid-point would need to be re-sent from the calling application. This is more likely to happen in a multi-part operation because those are longer, but a failure could conceivably happen during a single atomic operation as well.

With HA, if the library attempts to send a command to an HSM and it is unavailable, it will automatically retry sending that command to the next HSM in the configuration after the timeout expires.

Multi-part operations would typically be block encryption or decryption, or any other command where the previous state of the HSM is critical to the processing of the next command. It is understandable that these need to be re-sent since the HSMs do not synchronize ‘internal memory state,’ only stored key material.

Reaction to Failures

This section looks at possible failures in an overall HA system, and what needs to be done. The assumption is that HA has been In a complex system, it is possible to come up with any number of failure scenarios, such as this (partial) list for an HA group:

> Failure at the HSM or appliance
- HSM card failure
- HSM re-initialization
- Deactivated partition
- Power failure of a member
- Reboot of member
- NTL failure
- STC failure

> Failure at the client
- Power failure of the client
- Reboot of client
- Network keepalive failure

> Failure between client and group members
- Network failure near the member appliance
  (so only one member might disappear from client's view)
- Network failure near the client
  (client loses contact with all members)

HSM-Side Failures

The categories of failure at the HSM side of an HA arrangement are temporary or permanent.

Temporary

Temporary failures like reboots, or failures of power or network are self-correcting, and as long as you have set HA autorecovery parameters that are sufficiently lenient, then recovery is automatic, shortly after the HSM partition becomes visible to the HA client.

Permanent

Permanent failures require overt intervention at the HSM end, including possibly complete physical replacement of the unit, or at least initialization of the HSM.

All that concerns the HA service is that the particular unit is gone, and isn't coming back. If an entire SafeNet Luna Network HSM unit is replaced, then you must go through the entire appliance and HSM configuration of a new unit, before introducing it to the HA group. If a non-appliance HSM (resides in the Client host computer, e.g., SafeNet Luna PCIe HSM or SafeNet Luna USB HSM) is replaced, then it must be initialized and a new partition created.

Either way, your immediate options are to use a new name for the partition, or to make the HA SafeNet Luna HSM Client forget the dead member (LunaCM command **hagroup removemember**) so you can reuse the old name. Then, you must ensure that automatic synchronization is enabled (LunaCM command **hagroup synchronize -enable**), and manually introduce a new member to the group (LunaCM command **hagroup addmember**). After that, you can carry on using your application with full HA redundancy.

Because your application should be using only the HA virtual slot (LunaCM command **hagroup haonly**), your application should not have noticed that one HA group member went away, or that another one was added and synchronized. The only visible sign might have been a brief dip in performance, but only if your application was placing high demand on the HSM(s).

Client-Side Failures

For SafeNet Luna Network HSM, any failure of the client (such as operating system problems), that does not involve corruption or removal of files on the host, should resolve itself when the host computer is rebooted.

If the host seems to be working fine otherwise, but you have lost visibility of the HSMs in LunaCM or your client, verify that the SafeNet drivers are running, and retry. If that fails, reboot. If that fails, restore your configuration from backup of your host computer. If that fails, re-install SafeNet Luna HSM Client, re-perform certificate exchanges, creation of HA group, adding of members, setting HAOnly, etc.

For SafeNet Luna PCIe HSM and SafeNet Luna USB HSM, the client is the host of the HSMs, so if HA has been working, then any sudden failure is likely to be OS or driver related (so restart) or corruption of files (so re-install). If a re-install is necessary, you will need to recreate the HA group and re-add all members and re-assert all settings (like HAOnly).

Failures Between the HSM and Client (SafeNet Luna Network HSM only)

The only failure that could likely occur between a SafeNet Luna Network HSM (or multiple HSMs) and a client computer coordinating an HA group is a network failure. In that case, the salient factor is whether the failure occurred near the client or near one (or more) of the SafeNet Luna Network HSM appliances.

If the failure occurs near the client, and you have not set up port bonding on the client, then the client would lose sight of all HA group members, and the client application would fail. The application would resume according to its timeouts and error-handling capabilities, and HA would resume automatically if the members reappeared within the recovery window that you had set.

If the failure occurs near a SafeNet Luna Network HSM member of the HA group, then that member might disappear from the group until the network failure is cleared, but the client would still be able to see other members, and would carry on normally.

If the recovery window is exceeded, then you must manually restart HA.

Recovery

After a failure, the recovery process is typically straightforward. Depending on the deployment, an automated or manual recovery process might be appropriate. In either case there is no need to restart an application.

Automatic recovery

With automatic recovery, the client automatically performs periodic recovery attempts while a member is failed. The frequency of these checks is adjustable and the number of re-tries can be limited. Each time a reconnection is attempted, one application command experiences a slight delay while the client attempts to recover. As such, the retry frequency cannot be set any faster than once per minute. Even if a manual recovery process is selected, the application does not need to be restarted. 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.

Automatic recovery is disabled by default. Use the command **hagroup retry** to turn it on or off. If retry=0, automatic recovery is disabled. Any other retry value enables automatic recovery.

Failed units

Sometimes a failure of a device is permanent. In this event, the only solution is to deploy a new member to the group. In this case, you can remove the failed unit from the HA group, add a new device to the group and then start the recovery process. The running clients automatically resynchronize keys to the new member and start scheduling operations to it. See "Adding, Removing, Replacing, or Reconnecting HA Group Members" on page 155 for more information.

Manual recovery

Finally, sometimes both an HSM and application fail at the same time. If no new key material was created while an HSM was offline, the recovery is straightforward: simply return the HSM to service and then restart the application. However, if new key material was created after an HSM failed but before the application failed, a manual re-synchronization (using the **hagroup synchronize** command) might be required.

To perform a manual recovery, you confirm which member, or members, have the current key material (normally the unit that was online at the time the application failed). Put them back in service with the application. Then, for each member that has stale key material (a copy of an object that was deleted; or an old copy of an object whose attributes were changed), delete all their key material after making sure they are not part of the HA group. Be particularly careful that the member is not part of the HA group or the action might destroy active key material by causing an accidental synchronization during the delete operation. After the HSM is cleared of key material, rejoin it to the group and the synchronization logic automatically repopulates the device’s key material from the active units.

Usage

When a client is configured to use auto recovery the manual recovery commands must not be used. Invoking them can cause multiple concurrent recovery processes which result in error codes and possible key corruption .

Most customers should enable auto-recovery in all configurations. We anticipate that the only reason you might wish to choose manual recovery is if you do not want to change the retry time for periodic transactions. That is, each time a recovery is attempted a single application thread experiences an increased latency while the library uses that thread to attempt the re-connection (the latency impact is a few hundred milliseconds).

Recovery Conditions

HA recovery is hands-off resumption by failed HA Group members, or it is manual re-introduction of a failed member, if autorecovery is not enabled. Some reasons for a member to fail from the group might be:

- The appliance loses power (but regains power in less than the 2 hours that the HSM preserves its activation state).
- The network link from the unit is lost and then regained.

HA recovery takes place if the following conditions are true:

- HA autorecovery is enabled, or if you detect a unit failure and manually re-introduce the unit (or its replacement)
- HA group has at least 2 nodes
- HA node is reachable (connected) at client startup
- HA node recover retry limit is not reached. Otherwise manual recover is the only option to bring back the downed connection(s)

If all HA nodes fail (no links from client) no recovery is possible.

The HA recovery logic makes its first attempt at recovering a failed member when your application makes a call to its HSM (the HA group). An idle client does not start the recovery-attempt process. As of release 6.22, if the retry count is not 0, then recovery is attempted after the configured HA interval expires.

On the other hand, a busy client would notice a slight pause every minute, as the library attempts to recover a dropped HA group member (or members) until the member has been reinstated or until the timeout has been reached and it stops trying. Therefore, set the number of retries according to your normal situation (the kinds and durations of network interruptions you experience, for example).

Enabling and Configuring Autorecovery

In previous releases, autorecovery was not on by default, and needed to be explicitly enabled.

Beginning with SafeNet Luna HSM release 6.0, HA autorecovery is automatically enabled when you set the recovery retry count using the LunaCM command **hagroup retry**. Use the command **hagroup interval** to specify the interval, in seconds, between each retry attempt. The default is 60 seconds.

Failure of All Members

If all members of an HA group were to fail, then all logged-in sessions are gone, and operations that were active when the last group member went down, are terminated. If the client application is able to recover all that state information, then it is not necessary to restart or re-initialize in order to resume client operations with the SafeNet Luna Network HSM HA group. All sessions will be restarted without requiring a restart of the client.

Automatic Reintroduction

Automatic reintroduction is supported. A failed (and fixed, or replacement) HSM appliance can be re-introduced if the application continues without restart. Restarting the application causes it to take a fresh inventory of available HSMs, and to use only those HSMs within its HA group. You cannot reintroduce a SafeNet Luna Network HSM that was not in the group when the application started.

Auto-insert

Automatic reintroduction or "auto-insert" is supported. A failed (and fixed, or replacement) HSM appliance can be re-introduced if the application continues without restart. Restarting the application causes it to take a fresh inventory of available HSMs, and to use only those HSMs within its HA group. You cannot [re]introduce a SafeNet Luna Network HSM that was not in the group when the application started.

Auto-insert is now the default behavior (from Client 6.2.1 and later). [list below satisfies LHSM-31162]

1. A running client automatically detects SafeNet Luna Network HSM appliance insertion and removal to/from its configuration.
2. Connection to the new SafeNet Luna Network HSM appliance occurs only if the client HA configuration also has a new HA member or an HA member gone missing.
3. A running client does not automatically disconnect from the appliance that has been removed from its configuration until the appliance goes offline (for example, disconnected or powered down).
4. A running client uses the new HA member that is being added to the HA group configuration and does not require the client to restart to do so.
5. A running client stops attempting to use the removed HA member that is being revoked from the HA configuration and does not require the client to restart to do so.
6. When a new member is added to the HA group, entries similar to the following appear in the client HA Log:

   ```
   Mon Feb  1 11:06:55 2016 : [6619] HA group: 11079656446993 detected new member member:
   286668019649

   Mon Feb  1 11:07:25 2016 : [6619] HA group: 11079656446993 recovery attempt #1 succeeded for
   member: 286668019649
   ```

7. When a HA member is removed from the HA group, entries similar to the following appear in the client HA Log:

   ```
   Mon Feb  1 11:07:45 2016 : [6619] HA group: 11079656446993 member: 286668019649 revoked
   ```

8. When a new SafeNet Luna Network HSM appliance is registered with a client that has HA configured with “Active recovery mode”, entries similar to the following appear in the client HA Log:

   ```
   Sun Jan 31 21:01:52 2016 : [3820] HA subsystem detected new server : 192.20.11.175

   Sun Jan 31 21:01:56 2016 : [3820] HA subsystem server 192.20.11.175 connected
   ```

   Entries like these appear only if item 3, above, is true. [LHSM-31294]

9. When an existing SafeNet Luna Network HSM appliance is removed from client that has HA configured with “Active recovery mode”, entries similar to the following appear in the client HA Log:

   ```
   Tue Feb  2 15:45:12 2016 : [28001] HA subsystem detected removal of server : 192.20.11.86
   ```

Synchronization

Synchronization of token objects is a manual process using the **hagroup synchronize** command. Synchronization locates any object that exists on any one physical HSM partition (that is a member of the HA group), but not on all others, and replicates that object to any partitions (among the group) where it did not exist.

This is distinct from the replication that occurs when you create or delete an object on the HA virtual slot. Creation or deletion against the virtual slot causes that change to be immediately replicated to all connected members (addition or deletion).

Effect of PED Operations

PED operations block cryptographic operations, so that while a member of an HA group is performing a PED operation, it will 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.

Network failures

If network connectivity fails to one or more connected SafeNet Luna Network HSM appliances, the HA group will be restored automatically subject to timeouts and retries, as follows:

> While the client application is active, and one HA group member is connected and active, other members can automatically resume in the HA group as long as retries have not stopped.

> If all members fail or if the client does not have a network connection to at least one group member, then the client application must be restarted, unless you have **recoveryMode activeEnhanced** enabled.

Process interaction

Other events and processes interact at different levels and in different situations as described below.

> **NOTE** All references to NTLS also apply to STC. Both NTLS and STC provide secure client-appliance connections.

At the lowest communication level, the transport protocol (TCP) is responsible for making and operating the communication connection between client and appliance (whether HA is involved or not). For SafeNet Luna Network HSM, the default protocol timeout of 2 hours was much too long, so SafeNet configured that to 3 minutes when HA is involved. This means that:

> In a period of no activity by client or appliance, the appliance's TCP will wonder if the client is still there, and will send a packet after 3 minutes of silence.

> If that packet is acknowledged, the 3 minute TCP timer restarts, and the cycle repeats indefinitely.

> If the packet is not acknowledged, then TCP sends another after approximately 45 seconds, and then another after a further 45 seconds. At the two minute mark, with no response, the connection is considered dead, and higher levels are alerted to perform their cleanup.

So altogether, a total of five minutes can elapse since the last time the other participant was heard from. This is at the transport layer.

Above that level, the NTLS layer provides the connection security and some other housekeeping. 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 sends a "keep-alive PING" every two seconds, until the HSM returns the answer, which NTLS then conveys across the link to the requesting client. NTLS (nor any layer above) does not perform any interpretation of the ping.

It simply drops a slow, steady trickle of bytes into the pipe, to keep the TCP layer active. This normally has little effect, but if your client requests a lengthy operation like an 8192-bit keygen, then the random-number-generation portion of that operation could take many minutes to complete, 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.

Configuration settings

In the SafeNet configuration file, "DefaultTimeout" (default value is 500 seconds) governs how long the client will wait for a result from an HSM, for a cryptographic call. In the case of SafeNet Luna Network HSM, the copy of the config file inside the appliance is not accessible externally. The config file in the client installation is accessible to modify, but "DefaultTimeout" in that file affects only a locally connected HSM (such as might be the case if you had a SafeNet Luna Backup HSM attached to your client computer). The config file in the client has no effect on the configuration inside the network-attached SafeNet Luna Network HSM appliance, and thus can have no effect on the interaction between client and SafeNet Luna Network HSM appliance.

"ReceiveTimeout" is how long the library will wait for a dropped connection to come back.

If "ReceiveTimeout" is tripped, for a given appliance, the HA client stops talking to that appliance and deals with the remaining members of the HA group to serve your application's crypto requests.

A minute later, the HA client tries to contact the member that failed to reply.
If the connection is successfully re-established, the errant appliance resumes working in the group, being assigned application calls as needed (governed by application workload and HA logic).

If the connection is not successfully re-established, the client continues working with the remaining group members. Another minute passes, and the client once again tries the missing appliance to see if it is ready to actively resume working in the HA group.

The retries continue until the missing member resumes, or until the pre-set (by you) number of retries is reached (maximum of 500). If the retry count is reached with no success, the client stops trying that member. The failed appliance is still a member of the group (it is still in the list of HA group members maintained on the client), but the client no longer tries to send it application calls, and no longer encourages it to establish a connection. You must fix the appliance (or its network connection) and manually recover it into the group for the client to resume including it in operations.

Active Autorecovery on a SafeNet Luna Network HSM

> **NOTE** All references to NTLS also apply to STC. Both NTLS and STC provide secure client-appliance connections.

Autorecovery uses the HA Active Recovery Thread (ARCT) to manage recovery from a failure. The ARCT sends a non-session-based message that is processed by NTLS. This allows recovery as soon as a failed member returns. Thus, if a failed member returns to duty before an active member fails, then synchronization occurs immediately, and the secondary member is ready to take over from the active member if that now fails.

Members can reconnect without the need to call finalize/initialize in the client application, which allows for multiple services that use a single JVM to recover connections independently.

In the event that all HA members fail to respond to the ARCT probing message, the HA slot is deemed to be unrecoverable.

The recovery mode on a SafeNet Luna Network HSM is the basic active mode. As long as the retry count is not 0, recovery is active basic be default.

The enhanced active recovery mode is optional, and is controlled by the LunaCM **hagroup recoverymode** command.

Performance

For repetitive operations, like a high volume of signings using the same key, an HA group can expand SafeNet Luna Network HSM performance in linear fashion as HA group members are added. HA groups of 32 members have undergone long-term, full-throttle testing, with excellent results.

Do keep in mind that simply adding more and more SafeNet Luna Network HSM appliances to an HA group is not an infallible recipe for endless performance improvement. For best overall performance, all HA group members should be driven near their individual performance "sweet spot", around 30 simultaneous threads per HSM. If you assemble an HA group that is considerably larger than your server(s) can drive, then you might not achieve full performance from all.

The best approach is an HA group balanced in size for the capability of the application servers that will be driving the group, and the expected loads - with an additional unit to provide capacity for bursts of traffic and for redundancy.

Maximizing Performance

The SafeNet Luna Network HSM used in HA can provide performance improvement for asymmetric single-part operations. Gigabit Ethernet connections are recommended to maximize performance. For example, we have seen as much as a doubling of asymmetric single-part operations in a two-member group in a controlled laboratory environment (without crossing subnet boundaries, without competing traffic or other latency-inducing factors).

Multi-part operations are not load-balanced by the SafeNet HA due to the overhead that would be needed to perform context replication for each part of a multi-part operation.

Single-part cryptographic operations are load-balanced by the SafeNet HA functionality under most circumstances. Load-balancing these operations provides both scalability (better net throughput of operations) and redundancy by supporting transparent fail-over.

Performance is Dependent on the Type of Operation

Performance is also affected by the kind of operation you are performing. HA is better for performance when all HSM operations are performed on keys and material that reside within the HSM. This changes if part of the operation involves importing and unwrapping of keys; it can be instructive to consider what happens when such HSM operations are performed both with and without HA.

With HA

- One encryption (to wrap the key)
- One decryption in the HSM (to unwrap the key)
- Object creation on the HSM (the unwrapped key is created and stored as a key object)
- Key replication happens for HA

  - RSA 4096-bit operation used to derive a shared secret between HSM
  - Encryption of the key on the primary HA member using the shared secret
  - Decryption of the key on the secondary HA member HSM using the shared secret
  - Object creation on the second HA member
- One encryption (uses the unwrapped key object to encrypt the data)

Without HA

- One encryption (to wrap the key)
- One decryption in the HSM (to unwrap the key)
- Object creation on the HSM (the unwrapped key is created and stored as a key object)
- One encryption (uses the unwrapped key object to encrypt the data)

From the above it is apparent that, with HA, many more operations are performed. Most significant in the above case are the RSA 4096-bit operation and the additional object creation performed. Those two operations are by far the slowest operations in the list, and so this type of task would have much better performance without HA.

By contrast, if the task had made use of objects already within the HSM, then at most a single synchronization would have propagated the objects to all HA members, and all subsequent operations would have seen a performance boost from HA operation. The crucial consideration is whether the objects being manipulated are constant or are constantly being replaced.

HA and FindObjects

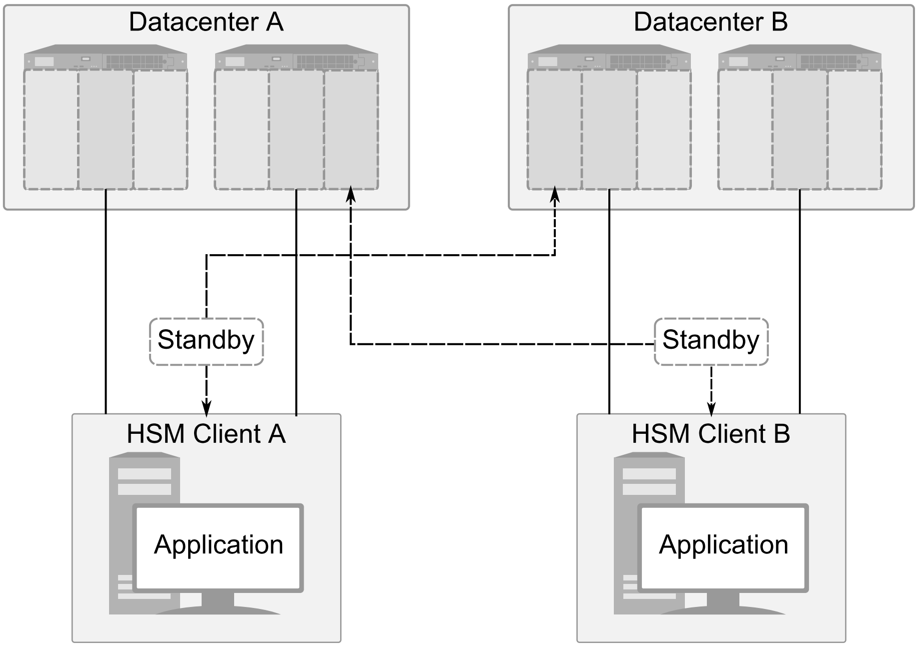
How your application uses the C_FindObjects function to search for objects in a virtual HA slot can have a significant impact your application performance. See "Application Object Handles" on page 154 for more information.

Standby Members

You can designate some members of an HA group as standby members after you add them to an HA group. Standby members differ from the default active members in that they do not actively participate in the HA group unless perform any cryptographic operations

By default, all members in an HA group are treated as active so that they are kept current with key material and are used to load-balance cryptographic services. In some deployment scenarios, however, it makes sense to define some members as standby. Standby members are registered just like active members except that they are defined as “standby” after they are added to the HA group.

As depicted below, applications can be deployed in geographically dispersed locations. In this scenario, you can use Luna’s standby capability to use the HSMs in the remote data center to cost-effectively improve availability. In this mode, only the local units (non-standby) are used for active load-balancing. However, as key material is created, it is automatically replicated to both the active (local) units and standby (remote) unit. In the event of a failure of all local members, the standby unit is automatically promoted to active status. You can use this feature to reduce costs, while improving reliability. This approach allows remote HSMs that have high latency to be avoided when not needed. However, in the worst case scenario where all the local HSMs fail, the remote member automatically activates itself and keeps the application running.



> **NOTE** In normal operation, the HA standby units do not perform any cryptographic operations. However, the HA service must log into all units in a group (C_OpenSession/Login is performed against all members), including standby units. This is necessary because, in the case where the standby unit is called into action, it must already be up-to-date with respect to key material that is being used in the group - it cannot synchronize with HSMs that have failed or that have gone off-line. Therefore, when the HA group consists of PED-authenticated HSMs, they must all be Activated, including the standby HSM(s).

Standby Behavior

Standby members become active only to keep the group alive. In an HA group that includes more than one standby member, if all active members go down/off-line, all available standby members become active in the group. Additional standby members remain on standby until/unless they are needed.

In other words, in an HA group, the load-sharing and redundancy capability is as large as all the active members. If all active members become unavailable to the application, then the group load-sharing and redundancy falls to all available standby members.

To set an HSM to standby status:

In "Configuring HA" on page 148, we created an HA group with label "myHAgroup" and group number 1154438865297, with two active members, serial number 154438865297 and serial number 1238700701520.

1. Create a third member, as previously described, and add it to the HA group by specifying either its slot or serial number.

   hagroup addmember -group <grouplabel> {**-slot** <slotnum> | **-serialnumber** <serialnum>}

   For example:

   ```
   lunacm:> hagroup addmember -group myHAgroup -slot 2

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

```
        Member 1238700701521 successfully added to group myHAgroup. New group
        configuration is:

         HA Group Label:  myHAgroup
        HA Group Number:  1154438865297
       HA Group Slot ID:  6
       Synchronization: enabled
          Group Members:  154438865297, 1238700701520, 1238700701521
             Needs sync:  no
        Standby Members:  <none>

Slot #    Member S/N                      Member Label    Status
======    ==========                      ============    ======
     0  154438865297                     HApartition00     alive
     1  1238700701520                     HApartition01     alive
     2  1238700701521                     HApartition02     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
```

2. Set the member to standby status, specifying its slot or serial number.

   hagroup addstandby -group <grouplabel> {**-slot** <slotnum> | **-serialnumber** <serialnum>}

   For example:

```
lunacm:> hagroup addstandby -group myHAgroup -serialnumber 1238700701521

        The member 1238700701521 was successfully added to the standby list for the HA Group
myHAgroup.

Command Result : No Error
```

3. If you wish, check the new configuration.

   hagroup listgroups

   For example:

```
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:  1154438865297
       HA Group Slot ID:  6
       Synchronization: enabled
```

```
             Group Members:  154438865297, 1238700701520, 1238700701521
                Needs sync:  no
           Standby Members:  1238700701521

   Slot #    Member S/N                      Member Label    Status
   ======    ==========                      ============    ======
        0  154438865297                     HApartition00     alive
        1  1238700701520                     HApartition01     alive
        2  1238700701521                     HApartition02     alive

   Command Result : No Error
```

Planning Your Deployment

This section describes the supported configurations and any limitations or constraints to consider when setting up an HA group.

HA Group Members

It is important that all members in an HA group have the same configuration and version. That means that each HA group member must use the same authentication method, either PED-authenticated or password-authenticated, and be at the same software version. Running HA groups with different versions is unsupported. Ensure that HSMs are configured identically to ensure smooth high availability and load balancing operation. SafeNet Luna HSMs come with various key management configurations: cloning mode, key-export mode, etc. HA functionality is supported with cloning, provided all members in the group have the same configuration. Clients automatically and transparently use the correct secure key replication method based on the group’s configuration.

It is also critical that all members in an HA group share the same Security Domain role (Red PED key for PED-authenticated devices, or domain password for password-authenticated devices). The Security Domain defines which HSMs are allowed to share key material. Because HA group members are, by definition, intended to be peers, they must be in the same Security Domain.

The SafeNet HA and load-balancing feature works on per-client and per-partition bases. This provides a lot of flexibility. For example, it is possible to define a different sub-set of HSMs in each client and even in each client’s partitions (in the event that a single client uses multiple partitions). SafeNet recommends to avoid these complex configurations and to keep the HA topography uniform for an entire HSM. That is, treat HSM members at the HSM level as atomic and whole. This simplifies the configuration management associated with the HA feature.

Mix and Match Appliance Software is Not Supported

All SafeNet Luna Network HSM appliances in an HA group must be running the same appliance software version. Before attempting to create an HA group, ensure that all of the appliances used to host the HA members are running the same appliance software. In addition, it is recommended that your client software is at the same software version as the appliance.

Mix and Match HSM Firmware, Capabilities, and FIPS Setting is Not Recommended

The HSM firmware, capabilities, and FIPS setting define which mechanisms are available, and how they can be used. To ensure that all objects in an HA slot can be successfully cloned to all members of the HA group, ensure that all members of a production HA group are at the same firmware level, have the same set of capabilities installed, and use the same FIPS setting. If mismatches exist between members, HSM operations or HA synchronization might fail if your application attempts to use a mechanism or a capability that not all members support.

To ensure minimal disruption during the during firmware or capability updates, your HA group will continue to function if there are differences in firmware, capabilities, or FIPS setting between the HA group members. Where differences exist, the capability of the group (in terms of features and available algorithms) is that of the member with the oldest firmware. It is recommended that you limit periods where mismatches are present to maintenance windows used to apply firmware of capability upgrades.

Example

Assume you have an HA group that includes HSMs with two different firmware versions,. In this case, certain capabilities that are part of the newer firmware are unavailable to clients connecting to the HA group. Specifically, operations that make use of newer cryptographic mechanisms and algorithms would likely fail. The client's calls might be initially assigned to a newer-firmware HSM and could therefore appear to work for a time, but if the task is load-balanced to an HSM that does not support the newer features, it would fail. Similarly, if the newer-firmware HSM dropped out of the group, operations requiring the newer firmware would fail.

HA Group Members Must Not Be on the Same Appliance

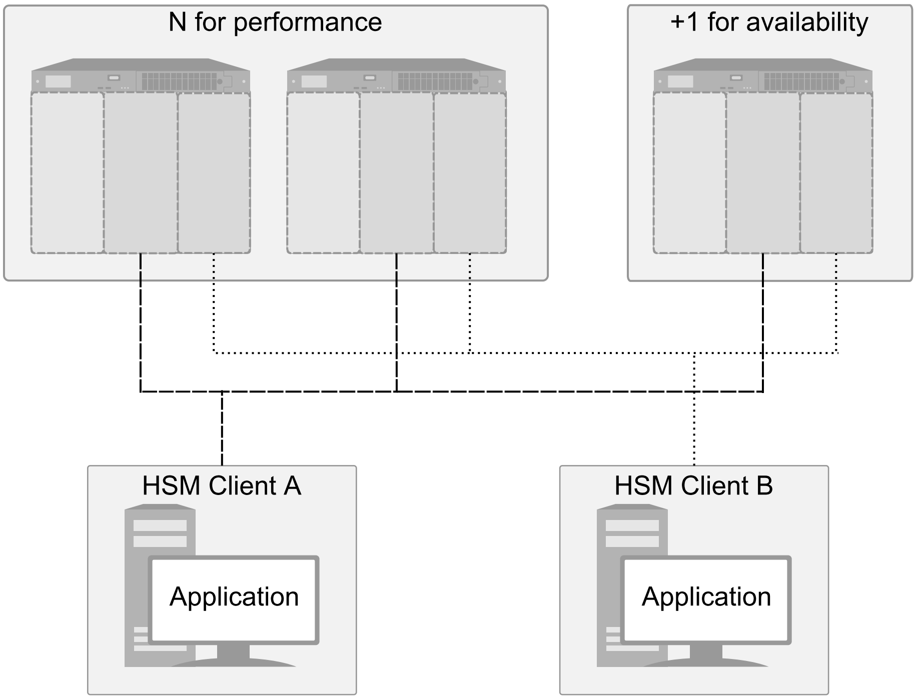
In any one HA group, always ensure that member partitions or member PKI tokens (USB-attached SafeNet Luna USB HSMs, or SafeNet CA4/PCM token HSMs in a USB-attached SafeNet DOCK2 card reader) are on different / separate appliances. Do not attempt to include more than one HSM partition or PKI token (nor one of each) from the same appliance in a single HA group. This is not a supported configuration. Allowing two partitions from one HSM, or a partition from the HSM and an attached HSM (as for PKI), into a single HA group would defeat the purpose of HA by making the SafeNet appliance a potential single-point-of-failure.

Running HA on a group of export SafeNet Luna Network HSM appliances

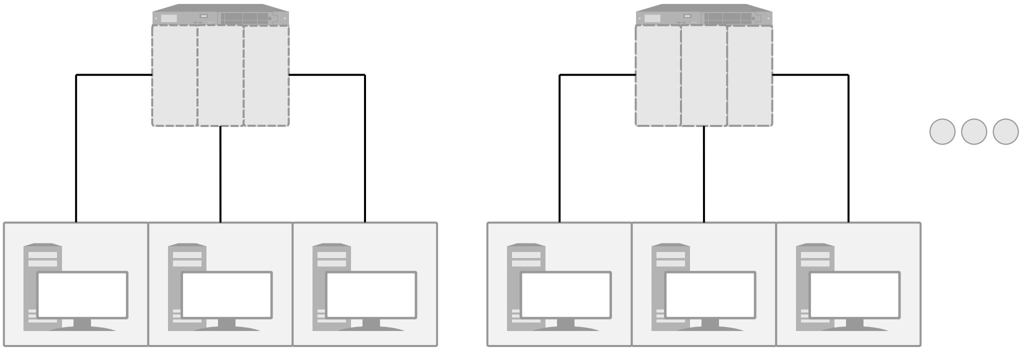
This configuration is supported, although you cannot clone/replicate private keys.

High Availability Group Sizing

As of SafeNet Luna HSM release 6.x, the high availability function supports the grouping of up to thirty-two members. However, the maximum practical group size for your application is driven by a trade-off between performance and the cost of replicating key material across the entire group. A common practice is to set the group size to N+1 where N is defined by the desired performance per application server(s). As depicted below, this solution gives the desired performance with a single extra HSM providing the availability requirement. The number of HSMs per group of application servers varies based on the application use case but, as depicted, groups of three are typical.



As performance needs grow beyond the performance capacity of three HSMs, it often makes sense to define a second independent group of application servers and HSMs to further isolate applications from any single point of failure. This has the added advantage of facilitating the distribution of HSM and application sets in different data centers.



Network Requirements

The network topography of the HA group is generally not important to the proper functioning of the group. As long as the client has a network path to each member the HA logic will function. Keep in mind that having a varying range of latencies between the client and each HA member causes a command scheduling bias towards the low-latency members. It also implies that commands scheduled on the long-latency devices have a larger 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 latencies to each device in the group are similar (with the exception of standby members, who do not contribute to network load). Gigabit Ethernet network connections are recommended.

Upgrading and Redundancy and Rotation

For SafeNet Luna Network HSM HA function we suggest that all SafeNet Luna Network HSM appliances in an HA group be at the same appliance software and firmware level. The issue is not about firmware level, per se - what might happen is that a newer firmware could contain newer algorithms that are not supported in the replaced firmware. If your client is configured to take advantage of newer/better algorithms when they become available, it might do so while one member of an HA group has new firmware, but another member has not yet been updated, and therefore does not yet support the requested algorithm. The client might not be able to interpret the resulting imbalance. Therefore, when you intend to upgrade/update any of the SafeNet Luna Network HSM units in an HA group, or when you intend to upgrade/update the SafeNet Luna Network HSM Client software, you might schedule some downtime for your application, if you anticipate a problem.

If the application is so critical that you cannot permit that much scheduled downtime, then you can set up a second complete set of Client computer and associated HA group. One set can service the application load while the other set is being upgraded or otherwise maintained. For such up-time-critical applications, you might already have such a backup set of Client-plus-HA-group that you would rotate in and out of service during regular maintenance windows.

Configuring HA

To create an HA group, you need at least two SafeNet Luna Network HSMs with PED Authentication, or two with Password Authentication. You cannot use Password -Authenticated and PED-Authenticated SafeNet Luna Network HSMs simultaneously in an HA group. This section describes how to set up an HA group with partitions on different HSMs. It consists of the following major steps:

> "Prerequisites" below

> "Create the HA Group" on the next page

> "Verification" on page 152

> "HA Standby Mode [Optional]" on page 153

Prerequisites

You must complete these procedures before setting up an HA group. The prerequisite steps are divided into tasks performed by different roles.

HSM SO Prerequisites

1. Perform the network setup on two or more SafeNet Luna Network HSM appliances (see "Configure the SafeNet Appliance for Your Network" on page 1 in the *Configuration Guide*).
2. Ensure that HSM policies **7: Allow Cloning** and **16: Allow Network Replication** are "on" (see "Set the HSM Policies" on page 1 in the *Configuration Guide*). If your HSMs do not have the cloning option, then they will use the Key Export functionality to backup to (and restore from) a file, rather than a hardware Backup token.
3. Initialize the HSMs (see "HSM Initialization" on page 165 in the *Configuration Guide*).
4. Create a partition on each SafeNet Luna Network HSM. They do not need to have the same label.
5. Allow one or more clients to access the partitions using NTLS or STC links (see "Enable the Client to Access a Partition" on page 1 in the *Configuration Guide*).

Partition SO Prerequisites

1. Ensure that all the partitions to be included in the HA group are visible in LunaCM (see "Enable the Client to Access a Partition" on page 1 in the *Configuration Guide*).
2. Initialize all the partitions to be included in the HA group (see "Configure Application Partitions" on page 1 in the *Configuration Guide*). The partitions do not need to have the same label, but they must be initialized with the same cloning domain:
   - PED-authenticated HSMs must share the same red domain PED key
   - Password-authenticated partitions must share the same domain string

   In this example, the partitions have been initialized as HApartition00 (SN 154438865297) and HApartition01 (SN 1238700701520).
3. [OPTIONAL] If you are setting up a PED-authenticated HA group, ensure that each Partition is Activated and AutoActivated (see "Activation and Auto-Activation on PED-Authenticated Partitions" on page 185), so that it can retain/resume its "Activate" (persistent login) state through any brief power failure or other interruption.
4. Initialize the Crypto Officer role on all the partitions.

   role init -name co

Crypto Officer Prerequisites

1. Login to each partition as Crypto Officer and change the initial primary credential (password or black PED key). Use the same Crypto Officer credential for each partition to be included in the HA group.

   role login -name co

   role changepw -name co
2. If you are setting up a PED-authenticated HA group, change the initial secondary credential (challenge password). Use the same challenge password for each partition to be included in the HA group.

   role login -name co

   role changepw -name co -oldpw <old_challenge> **-newpw** <new_challenge>

Create the HA Group

> **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 sufficient privileges.

After satisfying the prerequisites, use LunaCM to create an HA group on your client, and add member partitions. This procedure is completed by the Crypto Officer.

1. Use the **hagroup creategroup** command to create a new HA group on the client, which requires:
   - a Label for the group (do NOT call the group just "HA").
   - the Serial number OR the slot number of the primary partition.
   - the Crypto Officer password for the partition.

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

LunaCM generates and assigns a serial number to the group itself.

For example:

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

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

        New group with label "myHAgroup" created with group number 1154438865297.
        Group configuration is:

         HA Group Label:  myHAgroup
        HA Group Number:  1154438865297
       HA Group Slot ID:  Not Available
       Synchronization: enabled
          Group Members:  154438865297
             Needs sync:  no
        Standby Members:  <none>

Slot #    Member S/N                      Member Label    Status
======    ==========                      ============    ======
     0  154438865297                     HApartition00     alive

Command Result : No Error

LunaCM v7.0.0. Copyright (c) 2006-2017 SafeNet.

        Available HSMs:

        Slot Id ->              0
        Label ->                HApartition00
        Serial Number ->        154438865297
        Model ->                LunaSA 7.0.0
        Firmware Version ->     7.0.1
        Configuration ->        Luna User Partition With SO (PW) Signing With Cloning Mode
        Slot Description ->     Net Token Slot

        Slot Id ->              1
        Label ->                HApartition01
        Serial Number ->        1238700701520
        Model ->                LunaSA 7.0.0
        Firmware Version ->     7.0.1
        Configuration ->        Luna User Partition With SO (PW) Signing With Cloning Mode
        Slot Description ->     Net Token Slot

        Slot Id ->              5
        HSM Label ->            myHAgroup
        HSM Serial Number ->    1154438865297
        HSM Model ->            LunaVirtual
        HSM Firmware Version -> 7.0.1
        HSM Configuration ->    Luna Virtual HSM (PW) Signing With Cloning Mode
        HSM Status ->           N/A - HA Group


Current Slot Id: 0
```

> **NOTE** The example above was generated using Password-authenticated SafeNet Luna Network HSMs. For PED-authenticated HSMs, have a Luna PED connected, the partition already activated, and provide the partition challenge secret as the password (must be the same for all members).

2. Your chrystoki.conf/crystoki.ini file should now have a new section:

```
[VirtualToken]
VirtualToken00Label=myHAgroup
VirtualToken00SN=1154438865297
VirtualToken00Members=154438865297
```

> **CAUTION!** Never insert TAB characters into the chrystoki.ini (Windows) or crystoki.conf (UNIX) file.

3. Add another partition to the HA group (HApartition01 on sa40).

hagroup addmember -group <grouplabel> {**-slot** <slotnum> | **-serialnumber** <serialnum>}

For example:

```
lunacm:> hagroup addmember -group myHAgroup -slot 1

        Enter the password: ********
        Member 1238700701520 successfully added to group myHAgroup. New group
        configuration is:

         HA Group Label:  myHAgroup
        HA Group Number:  1154438865297
       HA Group Slot ID:  5
        Synchronization:  enabled
          Group Members:  154438865297, 1238700701520
             Needs sync:  no
        Standby Members:  <none>

Slot #    Member S/N                      Member Label    Status
======    ==========                      ============    ======
     0  154438865297                     HApartition00     alive
     1  1238700701520                     HApartition01     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
```

4. Check Chrystoki.conf/crystoki.ini again, the VirtualToken section should now look like this:

```
[VirtualToken]
VirtualToken00Label=myHAgroup
VirtualToken00SN=1154438865297
VirtualToken00Members=154438865297,1238700701520
```

5. Use the following command when you are ready to replicate data between/among all members of the HA group.

hagroup synchronize -group <grouplabel>

If you have additional members to add to the group, do this first to save time by avoiding multiple synchronizations. The 'synchronize' command replicates all objects on all partitions across all other partitions. As there are no objects on our newly-created partitions yet, we do not need to run this command.

> **NOTE** Do not use this command when recovering a group member that has failed (or was taken down for maintenance). Use the command **hagroup recover -group** <grouplabel>.

Verification

In LunaCM, we now have three slots available: two physical slots (a partition on each HSM) and a third virtual slot that points at both physical slots at once, via load balancing. To test your HA setup, perform the following steps:

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 rsakeygen -key 4096 -nodestroy -slots 5

   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 use **partition showinfo** on the two physical slots. Check the object count under "Partition Storage":

```
        Current Slot Id: 0

lunacm:> partition showinfo

...(clip)...

        Partition Storage:
                Total Storage Space:   325896
                Used Storage Space:    9480
                Free Storage Space:    316416
                Object Count:          206
                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:    9480
                Free Storage Space:    316416
                Object Count:          206
                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 slot 5

   partition login

   partition clear

If you are satisfied that your HA setup is working, you can begin using your application against the HA virtual slot ("myHAgroup" in the example). The virtual slot assignment will change depending on how many more application partitions are added to your client configuration. This will not matter to your application, which invokes the HA group label, not a particular slot number.

HA Standby Mode [Optional]

If you wish to add an additional partition that will be designated a standby member, and not a regular participant in the group, see "Standby Members" on page 142.

Using HA With Your Applications

This section describes how HA affects your applications, and describes the settings you can use and actions you can take to mitigate any performance or stability issues.

HAOnly

By default, the client lists both the physical slots and virtual slots for the HA group. Directing applications at the physical slots bypasses the high availability and load balancing functionality. An application must be directed at the virtual slots to activate the high availability and load balancing functionality. A configuration setting referred to as **HAonly** hides the physical slots, and is recommended to prevent incorrect application configurations. Doing so also simplifies the PKCS #11 slot ordering given a dynamic HA group.

What is the impact of the ‘haonly’ flag, and why might you wish to use it?

The “haonly” flag shows only HA slots (virtual slots) to the client applications. It does not show the physical slots. We recommend that you use "haonly", unless you have particular reason for not using it. Having "haonly" set is the proper way for clients to deal with HA groups - it prevents the possible confusion of having both physical and virtual slots available.

Recall that automatic replication/synchronization across the group occurs only if you cause a change (keygen or other addition, or a deletion) via the virtual HA slot. If you/your application changes the content of a physical slot, this results in the group being out-of-sync, and requires a manual re-sync to replicate a new object across all members. Similarly, if you delete from a physical slot directly, the next manual synchronization will cause the deleted object to be repopulated from another group member where that object was never deleted. Therefore, to perform a lasting deletion from a single physical slot (if you choose not to do it via the virtual slot) requires that you manually delete from every physical slot in the group, or risk your deleted object coming back.

Also, from the perspective of the Client, a member of the HA group can fail and, with "haonly" set, the slot count does not change. If "haonly" is not set, and both virtual and physical slots are visible, then failure of unit number 1 causes unit number 2 to become slot 1, and so on. That could cause problems if your application is not designed to deal gracefully with such a change.

Key Generation

An application that continuously generates key material will need to have its HA group carefully selected since the generated session objects need to be replicated to each member of the HA group, requiring significant processing overhead that does not exist in single slot mode. Contact Technical Support for help in architecting your HA group for this type of application.

Application Object Handles

Application developers should be aware that the PKCS #11 object handle model is fully virtualized when using an HA slot. As such, 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

Because the client must perform a look up to create the virtual object table, how 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 trh 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.

Adding, Removing, Replacing, or Reconnecting HA Group Members

This section describes how to add a new member to an HA group, reconnect an offline member, or replace a failed unit.

Adding or Removing an HA Group Member

Use the following LunaCM commands to add or remove a normal or standby member to or from an HA group:

- **hagroup addmember**
- **hagroup addstandby**
- **hagroup removemember**
- **hagroup removestandby**

See "hagroup" on page 1 in the *LunaCM Command Reference Guide* for detailed descriptions and syntax for each **hagroup** command.

> **NOTE** You must restart the application to have the added or removed member recognized.

Reconnecting an Offline Unit

In HA mode, if an HSM appliance goes off-line or drops-out (due to failure, maintenance, or some other reason), the application load is spread over the remaining members of the HA group. When the appliance is restarted, the application does not need to be stopped and restarted before the re-introduced appliance can be used by the application. For the unit that was withdrawn (or for a replacement unit), if it was powered off for more than a short outage, you must re-activate the partitions before they can be re-included into the HA Group.

The following reconnection scenarios are available:

To recover the same group member

1. Restart the failed member and verify that it has started properly.
2. Do not perform a manual re-synchronization between the members. Instead, use the following LunaCM command:

   hagroup recover -group <group_name>

Replacing a Failed SafeNet Luna Network HSM

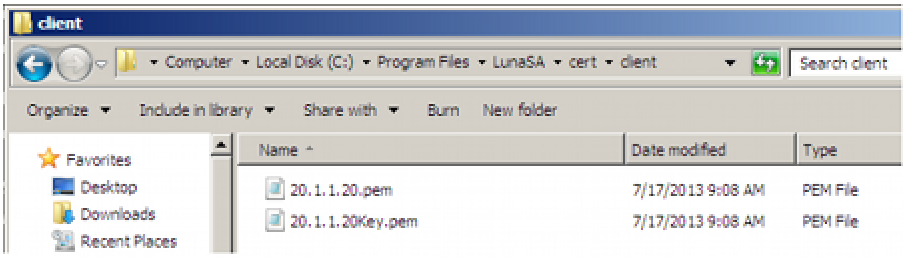
Before getting into replacing HSMs in an HA group, this first section describes relevant system conditions and settings to have a SafeNet Luna Network HSM configured and in an authenticated relationship with a client computer. In particular, we are interested in the client-side config file and the client's certificate folder in ordinary, single-appliance mode, and then in HA. You would already have set up the a SafeNet Luna Network HSM as described in the configuration manual, for network setup and creation of the appliance-side certificate (see "Generate a New HSM Server Certificate").

Chrystoki.ini before client-side certificate creation

```
[Chrystoki2]
LibNT=C:\Program Files\SafeNet\LunaClient\cryptoki.dll

[LunaSA Client]
SSLConfigFile=C:\Program Files\SafeNet\LunaClient\openssl.cnf
ReceiveTimeout=20000
NetClient=1
ServerCAFile=C:\Program Files\SafeNet\LunaClient\cert\server\CAFile.pem
ClientCertFile=C:\Program Files\SafeNet\LunaClient\cert\client\ClientNameCert.pem
ClientPrivKeyFile=C:\Program Files\SafeNet\LunaClient\cert\client\ClientNameKey.pem

[Luna]
DefaultTimeOut=500000
PEDTimeout1=100000
PEDTimeout2=200000
PEDTimeout3=10000

[CardReader]
RemoteCommand=1
```

1. Create client-side certs (see "vtl createCert " on page 1 in the *Utilities Reference Guide*).

   Generated client certificates



Name	Date modified	Type
20.1.1.20.pem	7/17/2013 9:08 AM	PEM File
20.1.1.20Key.pem	7/17/2013 9:08 AM	PEM File

   Chrystoki.ini after client-side certificate creation

```
[Chrystoki2]
LibNT=C:\Program Files\SafeNet\LunaClient\cryptoki.dll

[LunaSA Client]
SSLConfigFile=C:\Program Files\SafeNet\LunaClient\openssl.cnf
ReceiveTimeout=20000
NetClient=1
ServerCAFile=C:\Program Files\SafeNet\LunaClient\cert\server\CAFile.pem
ClientCertFile=C:\Program Files\SafeNet\LunaClient\cert\client\20.1.1.20.pem
ClientPrivKeyFile=C:\Program Files\SafeNet\LunaClient\cert\client\20.1.1.20.pem
```

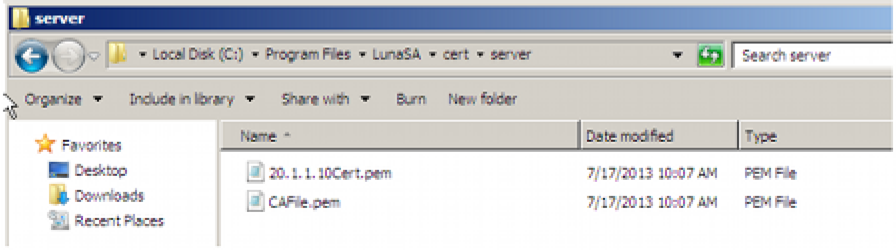
2. Copy the SafeNet Luna Network HSM **server.pem** to the client.

> **NOTE** At this point there are still no certificates in the **cert\server** directory.

3. Use **vtl addserver** to register the SafeNet Luna Network HSM with the client.

   CAFile.pem is generated in the **cert\server** directory.

Cert\server directory after CAFile.pem is generated



Name	Date modified	Type
20.1.1.10Cert.pem	7/17/2013 10:07 AM	PEM File
CAFile.pem	7/17/2013 10:07 AM	PEM File

Crystoki.ini after "vtl addserver"

```
[Chrystoki2]
LibNT=C:\Program Files\SafeNet\LunaClient\cryptoki.dll

[LunaSA Client]
SSLConfigFile=C:\Program Files\SafeNet\LunaClient\openssl.cnf
ReceiveTimeout=20000
NetClient=1
ServerCAFile=C:\Program Files\SafeNet\LunaClient\cert\server\CAFile.pem
ClientCertFile=C:\Program Files\SafeNet\LunaClient\cert\client\20.1.1.20.pem
ClientPrivKeyFile=C:\Program Files\SafeNet\LunaClient\cert\client\20.1.1.20.pem
ServerName00=20.1.1.20
ServerPort00=1792
```

vtl verify results

```
C:\Program Files\SafeNet\LunaClient>vtl verify

The following SafeNet Luna Network HSM Slots/Partitions were found:

Slot   Serial #        Label
====   ========        =====
1      154702010       p1

C:\Program Files\SafeNet\LunaClient>
```

Replace a SafeNet Luna Network HSM Using the Same IP

For an existing HA group, bring in a replacement SafeNet Luna Network HSM.

1. Change the IP of the new appliance to match the one that was removed.
2. Perform RegenCert on the new SafeNet Luna Network HSM.

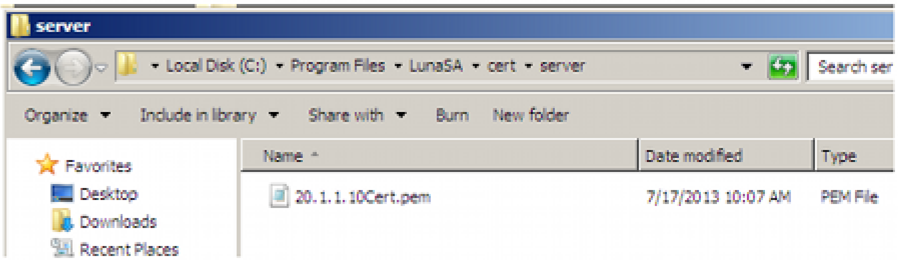
> **NOTE** **vtl verify** on client at this time would fail because the cert that the client has is for the old, removed SafeNet Luna Network HSM.

3. Execute **vtl deleteserver –n <original IP>**

   Deleting old SafeNet Luna Network HSM from Client

```
C:\Program Files\SafeNet\LunaClient>vtl listservers
Server: 20.1.1.20

C:\Program Files\SafeNet\LunaClient>vtl deleteserver -n 20.1.1.20
Server: 20.1.1.20 successfully removed from server list.

C:\Program Files\SafeNet\LunaClient>
```

   Contents of cert\server after “deleteserver” (CAFile.pem has been deleted)



Name	Date modified	Type
20.1.1.10Cert.pem	7/17/2013 10:07 AM	PEM File

4. Copy new **server.pem** to client.

   Copying new server.pem to client

```
C:\Program Files\SafeNet\LunaClient>pscp admin@20.1.1.20:server.pem .
admin@20.1.1.20's password:
server.pem              | 1 kB | 1.1 kB/s | ETA: 00:00:00 | 100%
```

5. Run **vtl addserver** using new **server.pem**

   vtl addserver using new server.pem

```
C:\Program Files\SafeNet\LunaClient>vtl addserver -n 20.1.1.20 -c server.pem
New server: 20.1.1.20 successfully added to server list.
```

6. Run **vtl verify**.

   vtl verify results

```
C:\Program Files\SafeNet\LunaClient>vtl verify

The following SafeNet Luna Network HSM Slots/Partitions were found:
```

```
Slot   Serial #        Label
====   ========        =====
1      154702010       p1
```

Summary

If a SafeNet Luna Network HSM must be replaced, the old IP can be used, but the SafeNet Luna Network HSM certificate must be regenerated. The IP must be removed from the server list on the client and then added back using the new **server.pem**.

Client side requirements review:

- Use **vtl deleteserver** to remove IP from list and delete CAFile.pem from **cert\server**.
- Copy new **server.pem** to client
- Use **vtl addserver** to re-add IP and create **CAFile.pem**.

Client-side - Reconfigure HA If a SafeNet Luna Network HSM Must Be Replaced

1. Note HA partition serial numbers

```
C:\Program Files\SafeNet\LunaClient>vtl verify
The following SafeNet Luna Network HSM Slots/Partitions were found:
Slot   Serial #        Label
====   ========        =====
1      154702011       HA1
1      154702012       HA2
```

2. Run **hagroup creategroup -serialnumber -password** with lunacm:>

   A group is created with HA1 as Primary.

```
lunacm:>hagroup creategroup -serialnumber 154702011 -label SomeHAGrp -password PassWd

Command Result: No Error
```

Crystoki.ini after HA group is created

```
[Chrystoki2]
LibNT=C:\Program Files\SafeNet\LunaClient\cryptoki.dll

[LunaSA Client]
SSLConfigFile=C:\Program Files\SafeNet\LunaClient\openssl.cnf
ReceiveTimeout=20000
NetClient=1
ServerCAFile=C:\Program Files\SafeNet\LunaClient\cert\server\CAFile.pem
ClientCertFile=C:\Program Files\SafeNet\LunaClient\cert\client\20.1.1.20.pem
ClientPrivKeyFile=C:\Program Files\SafeNet\LunaClient\cert\client\20.1.1.20.pem
ServerName00=20.1.1.20
ServerPort00=1792

[Luna]
DefaultTimeOut=500000
PEDTimeout1=100000
PEDTimeout2=200000
PEDTimeout3=10000
```

```
[CardReader]
RemoteCommand=1

[VirtualToken]
VirtualToken00Label=SomeHAGrp
VirtualToken00SN=1154702011
VirtualToken00Members=154702011

[HASynchronize]
SomeHAGrp=1
```

3. Add a secondary SafeNet Luna Network HSM partition to the HA group with lunacm:> **hagroup addmember -serialnumber -group -password**.

```
lunacm:> hagroup addmember -serialnumber 154702012 -group SomeHAGrp -password PassWd

Command Result: No Error
```

Crystoki.ini after second HA member is added

```
[Chrystoki2]
LibNT=C:\Program Files\SafeNet\LunaClient\cryptoki.dll

[LunaSA Client]
SSLConfigFile=C:\Program Files\SafeNet\LunaClient\openssl.cnf
ReceiveTimeout=20000
NetClient=1
ServerCAFile=C:\Program Files\SafeNet\LunaClient\cert\server\CAFile.pem
ClientCertFile=C:\Program Files\SafeNet\LunaClient\cert\client\20.1.1.20.pem
ClientPrivKeyFile=C:\Program Files\SafeNet\LunaClient\cert\client\20.1.1.20.pem
ServerName00=20.1.1.20
ServerPort00=1792

[Luna]
DefaultTimeOut=500000
PEDTimeout1=100000
PEDTimeout2=200000
PEDTimeout3=10000

[CardReader]
RemoteCommand=1

[VirtualToken]
VirtualToken00Label=SomeHAGrp
VirtualToken00SN=1154702011
VirtualToken00Members=154702011, 154702012

[HASynchronize]
SomeHAGrp=1
```

Crystoki.ini after HA Only is enabled

```
[Chrystoki2]
LibNT=C:\Program Files\SafeNet\LunaClient\cryptoki.dll

[LunaSA Client]
SSLConfigFile=C:\Program Files\SafeNet\LunaClient\openssl.cnf
ReceiveTimeout=20000
```

```
NetClient=1
ServerCAFile=C:\Program Files\SafeNet\LunaClient\cert\server\CAFile.pem
ClientCertFile=C:\Program Files\SafeNet\LunaClient\cert\client\20.1.1.20.pem
ClientPrivKeyFile=C:\Program Files\SafeNet\LunaClient\cert\client\20.1.1.20.pem
ServerName00=20.1.1.20
ServerPort00=1792

[Luna]
DefaultTimeOut=500000
PEDTimeout1=100000
PEDTimeout2=200000
PEDTimeout3=10000

[CardReader]
RemoteCommand=1

[VirtualToken]
VirtualToken00Label=SomeHAGrp
VirtualToken00SN=1154702011
VirtualToken00Members=154702011, 154702012

[HASynchronize]
SomeHAGrp=1

[HAConfiguration]
HAOnly=1
```

Crystoki.ini after "autorecovery" is enabled

```
[Chrystoki2]
LibNT=C:\Program Files\SafeNet\LunaClient\cryptoki.dll

[LunaSA Client]
SSLConfigFile=C:\Program Files\SafeNet\LunaClient\openssl.cnf
ReceiveTimeout=20000
NetClient=1
ServerCAFile=C:\Program Files\SafeNet\LunaClient\cert\server\CAFile.pem
ClientCertFile=C:\Program Files\SafeNet\LunaClient\cert\client\20.1.1.20.pem
ClientPrivKeyFile=C:\Program Files\SafeNet\LunaClient\cert\client\20.1.1.20.pem
ServerName00=20.1.1.20
ServerPort00=1792

[Luna]
DefaultTimeOut=500000
PEDTimeout1=100000
PEDTimeout2=200000
PEDTimeout3=10000

[CardReader]
RemoteCommand=1

[VirtualToken]
VirtualToken00Label=SomeHAGrp
VirtualToken00SN=1154702011
VirtualToken00Members=154702011, 154702012

[HASynchronize]
SomeHAGrp=1
```

```
[HAConfiguration]
HAOnly=1
reconnAtt=500
```

4. Show HA configuration results with **hagroup listgroups** in lunacm:>

```
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: 65 seconds
HA logging: enabled
HA log _file: /luna_ha_temp/haErrorLog.txt
Maximum HA log file length: 300000 bytes
Only Show HA Slots: no

HA Group Label: SomeHAGrp
HA Group Number: 1364882803566
HA Group Slot ID: 5
Synchronization: enabled
Group Members: 154702011, 154702012
Needs sync: no
Standby Members: <none>
Slot #  Member S/N Member Label Status
======  ========== ============ ======
0        154702011         HA1  alive
1        154702012         HA2  alive

Command Result : No Error
```

Replacing the Secondary HA Group Member

When the SafeNet Luna Network HSM to be replaced, in an HA Group, is a secondary member, the process is similar to above. You must delete the secondary from the HA Group and re-add it with the new partition serial number. It is not necessary to delete and recreate the group.

If a SafeNet Luna Network HSM must be replaced, the old IP address can be used, but the SafeNet Luna Network HSM certificate must be regenerated. The IP address must be removed from the server list on the client and then added back using the new “server.pem” received from the replacement SafeNet Luna Network HSM.

If the SafeNet Luna Network HSM being replaced is the Primary, you must delete the HA Group and recreate it using the new Primary SafeNet Luna Network HSM partition serial number and then add the original Secondary SafeNet Luna Network HSM partition serial number - the cert from the original Secondary is already in place on the client, and no change is needed to that.

Managing and Troubleshooting Your HA Groups

You can use vtl and the LunaCM **hagroup** commands to monitor and manage your HA groups.

Slot Enumeration

The client-side utility command **vtl listslot** or the LunaCM **slot list** command shows all detected slots, including HSM partitions on the primary HSM, partitions on connected external HSMs, and HA virtual slots. Here is an example:

bash-3.2# ./vtl listslot

Number of slots: 11

The following slots were found:

```
Slot #          Description             Label           Serial #        Status
slot #1         LunaNet Slot            -               -               Not present
slot #2         LunaNet Slot            sa76_p1         150518006       Present
slot #3         LunaNet Slot            sa77_p1         150475010       Present
slot #4         LunaNet Slot            G5179           700179008       Present
slot #5         LunaNet Slot            pki1            700180008       Present
slot #6         LunaNet Slot            CA4223          300223001       Present
slot #7         LunaNet Slot            CA4129          300129001       Present
slot #8         HA Virtual Card Slot    -               -               Not present
slot #9         HA Virtual Card Slot    -               -               Not present
slot #10        HA Virtual Card Slot            ha3     343610292       Present
slot #11        HA Virtual Card Slot            G5_HA   1700179008      Present
```

> **NOTE** - The deploy/undeploy of a PKI device increments/decrements the SafeNet Luna Network HSM client slot enumeration list (slots appear or disappear from the list, and the slot numbers adjust for the change). HA group virtual slots always appear toward the end of the list, following the physical slots. The actual slot number can vary based on the currently connected external HSMs (tokens, G5).

Due to the above behavior, we generally recommend that you run the lunacm:> **haGroup haonly** command so that only the HA slot is visible and any confusion or improper slot use is eliminated.

Determining Which Device is in Use

Use the **ntls show** or **stc status** command.

Determining Which Devices are Active

CA extension call “CA_GetHAState” lists all active devices. The LunaCM **hagroup listgroup** command also lists members.

Duplicate Objects

If you create an object on your HA slot, and then duplicate that object in some fashion (for example, by SIM'ing [wrapping] it off and then back on again, or performing a backup/restore with the 'add' option), that object will be seen as only one object on the HA slot because HA uses the object's fingerprint to build an object list. Two objects will in fact exist on each of the physical slots and could be seen by a non-HA utility/query to the HSM.

There are TWO implications from this situation:

- One implication is that repeated duplication (perhaps an application that performs periodic backups, and restores using the 'add' option rather than 'replace') could cause the partition to reach the maximum number of partition objects while seemingly having fewer objects. If the system ever tells you that your partition is full, but HA says otherwise, then use a tool like CKDemo that can view the "physical" slots directly (as opposed to the HA slot) on the HSM, and delete any objects that are unnecessary.
- A second implication is that the HA feature uses object fingerprints to match different instances of an object on different physical HSMs. This can result in error messages if your application does not properly create and destroy session objects, and perhaps creates an object identical to one which has been removed in a separate concurrent session. The problem is self-correcting, but the flurry of error messages could be worrying if you don't understand where they are coming from.

Frequently Asked Questions

This section provides additional information by answering questions that are frequently asked by our customers.

Can we manage NTLS connections through a load balancer (like NetScaler, Barracuda, A10, etc.)?

No. NTLS will not work through a load-balancer because it is an end-to-end TLS pipe between client and SafeNet Luna Network HSM.

We want to use a backup application server that would operate in standby mode until awakened by a failure of our primary application server. Can we use a virtual IP in the SafeNet Luna Network HSM setup, so that both primary and secondary are accepted for NTLS as the same client by SafeNet Luna Network HSM?

Yes. At the client, generate the client cert with the command **vtl createCert -n <any IP address, real or virtual>**.

Both client computers must have the SafeNet Luna Network HSM appliance's server cert in their client-side server-cert folders.

The SafeNet Luna Network HSM appliance must have the client certificate (built with the virtual IP address)

Also the following lines in the Chrystoki.conf file must point to the same cert and Keyfile on the clustered application servers:

```
LunaSA Client ={
 ClientCertFile=\usr\LunaClient\cert\client\<your-cert-filename>.pem
 ClientPrivKeyFile=\usr\LunaClient\cert\client\<your-filename>Key.pem
```

Our application keeps the HSM full. Can we double the capacity by creating an HA group and having a second HSM?

No. HA provides redundancy and can increase performance, but not capacity. Every HSM in an HA group gets synchronized with the other member(s), which means that the content of any one HSM in an HA group must be a clone of the content of any other member of that group. So, with more HA group members, you get more copies, not more space.

CHAPTER 8: 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 168.

> **NOTE** To ensure accurate auditing, perform initialization only after you have set the system time parameters (time, date, timezone, 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:

<table>
<tr><td><b>HSM Label</b></td><td>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.</td></tr>
<tr><td><b>HSM
SO credentials</b></td><td>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 194.
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 !#$%'*+,-./0123456789:=?@ABCDEFGHIJKLMNOPQRSTUVWXYZ []^_abcdefghijklmnopqrstuvwxyz{}~ (the first character in that list is the space character)
> Invalid characters are "&';<>\`|()</td></tr>
<tr><td><b>Cloning domain for the HSM Admin partition</b></td><td>The cloning domain is a shared identifier that makes cloning possible among a group of HSM partitions. It specifies the security domain (group of HSM partitions) within which the HSM Admin partition can share cryptographic objects though cloning, backup/restore, or in high availability configurations. Note that the HSM Admin partition cloning domain is independent of the cloning domain specified when creating application partitions on the HSM.
For PED-authenticated HSMs, you create a new Domain (red) PED key(set) or re-use an existing key(set) from an HSM you want to be able to clone with.
For password-authenticated HSMs, you create a new domain password or re-use an existing password from an HSM you want to be able to clone with. Cloning domain strings can be between 1 and 128 characters in length:
> Valid characters are !#$%'*+,-./0123456789:=?@ABCDEFGHIJKLMNOPQRSTUVWXYZ []^_abcdefghijklmnopqrstuvwxyz{}~ (the first character in that list is the space character)
> Invalid characters are "&';<>\`|()
<b>NOTE</b> 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.</td></tr>
</table>

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 275 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:

      lunash:> **hsm 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 202 in the *HSM Administration Guide*.

3. Log into LunaSH as the appliance administrator 'admin'. You can use a serial terminal window or SSH connection.

4. Run the **hsm init** command, specifying a label for your SafeNet Luna Network HSM:

   lunash:> **hsm init** -label <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 keysets and duplicate keys as required. See "PED Authentication" on page 194 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 174

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 166.

> **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 275 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 202 in the *HSM Administration Guide*.
4. Log into LunaSH as the appliance administrator 'admin'. You can use a serial terminal window or SSH connection.
5. Run the **hsm init** command, specifying a label for your SafeNet Luna Network HSM:

   lunash:> **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:

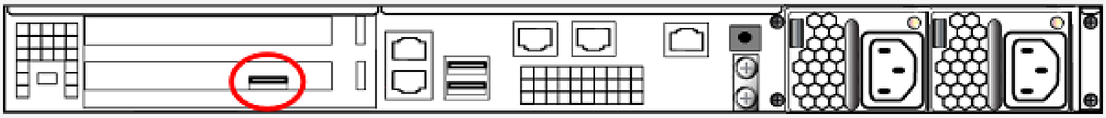
- "Detailed procedure" on the next page
- "Imprinting the Blue HSM SO PED Key" on page 170
- "Imprinting the Red Cloning Domain PED Key" on page 172
- "New, reuse, and overwrite options" on page 173

> **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.

Detailed procedure

1. Your Luna PED must be connected to the HSM, either locally/directly in USB mode (see "Changing Modes" on page 202), or remotely via Remote PED connection (see "About Remote PED" on page 204).

> **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 appliance.



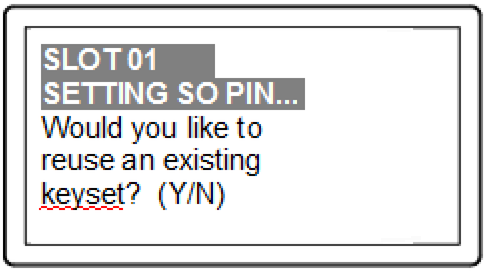
2. Set the active slot to the SafeNet Luna Network HSM Admin partition, and issue the **hsm init** command. The HSM passes control to the Luna PED, and the command line directs you to attend to the PED prompts.
3. When you issue the **hsm init** command, the HSM passes control to the Luna PED, and the 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 access control for your HSM (See "M of N Split Secrets" on page 199 for details).
10. Luna PED asks if you wish to use a PED PIN (an additional secret; see "PED Key Management" on page 224 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 LunaSH.

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

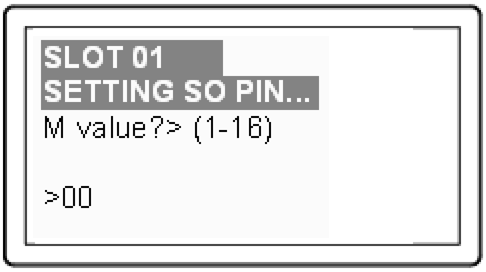
1. Decide if you want to reuse a keyset.



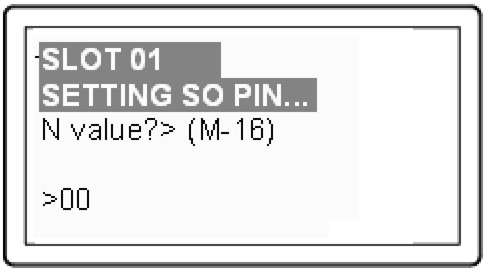
```
SLOT 01
SETTING SO PIN...
Would you like to
reuse an existing
keyset?  (Y/N)
```

- 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.



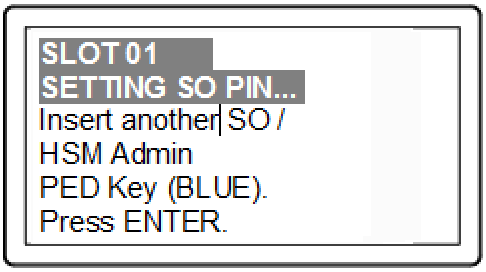
```
SLOT 01
SETTING SO PIN...
M value?> (1-16)

>00
```



```
SLOT 01
SETTING SO PIN...
N value?> (M-16)

>00
```

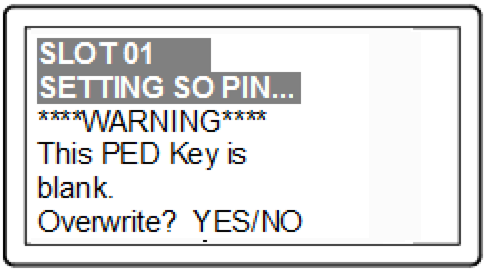
- Setting M and N to **1** means that the authentication is not to be split, and only a single PED key will be necessary when the authentication is called for in future. Input **1** for each prompt if you do not want to use MofN.
- Setting M and N to larger than 1 means that the authentication is split into N different splits, of which quantity M of them 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 other holders.

3. Insert your blank key or the key you wish to overwrite.



```
SLOT 01
SETTING SO PIN...
Insert another SO /
HSM Admin
PED Key (BLUE).
Press ENTER.
```

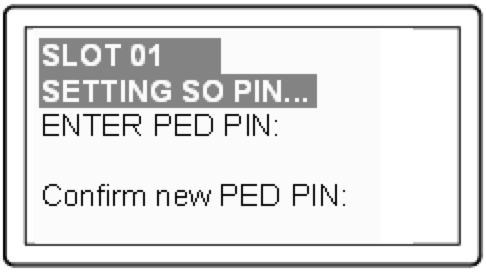
Insert a blue HSM Admin/SO PED key and press **Enter**.



```
SLOT 01
SETTING SO PIN...
****WARNING****
This PED Key is
blank.
Overwrite?  YES/NO
```

- **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.

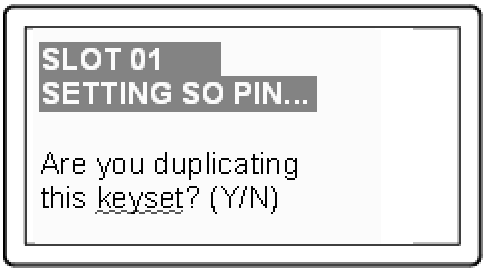
4. For any situation other than reusing a keyset, Luna PED now prompts for you to set a PED PIN. For multi-factor authentication security, the physical PED key is "something you have." You can choose to associate that with "something you know," in the form of a multi-digit PIN code that must always be supplied along with the PED key for all future HSM access attempts.



```
SLOT 01
SETTING SO PIN...
ENTER PED PIN:

Confirm new PED PIN:
```

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.



SLOT 01
SETTING SO PIN...

Are you duplicating
this keyset? (Y/N)

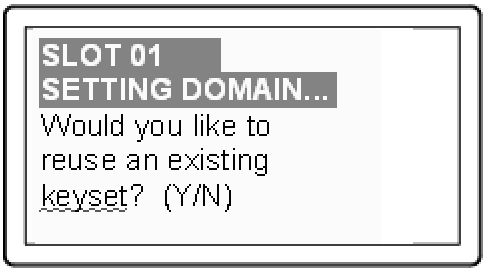
- **Yes**: Present one or more blank keys, all of which will be imprinted with exact copies of the current PED key's authentication.
- **No**: Do not make any copies.

> **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.



SLOT 01
SETTING DOMAIN...
Would you like to
reuse an existing
keyset? (Y/N)

- **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.
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 198) 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

```
lunash:>hsm init -label myLunaHSM

  Please enter a password for the HSM Administrator:
  > ********

  Please re-enter password to confirm:
  > ********

  Please enter a cloning domain to use for initializing this HSM:
  > *********

  Please re-enter cloning domain to confirm:
  > *********

CAUTION:  Are you sure you wish to initialize this HSM?

          Type 'proceed' to initialize the HSM, or 'quit'
          to quit now.
          > proceed

'hsm init' successful.

Command Result : 0 (Success)
```

When activity is complete, the system displays a “success” message.

CHAPTER 9:
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. In LunaSH, this information is displayed under *HSM Details* by running **hsm show**.

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 165 for more information.		

For a comparison of various destruction or denial actions on the HSM, see "Comparison of Destruction/Denial Actions" on page 124.

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

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 "Create Application Partitions" 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.

Depending upon the configuration, each SafeNet Luna Network HSM can contain a number of HSM Partitions (according to your license agreement). Each HSM Partition has the capacity to hold data objects in numbers that depend upon the memory available, divided among number of partitions that your HSM allows. The HSM SO can use the LunaSH **partition resize** command to modify the sizes of individual partitions until all memory on the HSM is allotted to, for example, make room for some larger partitions by shrinking others.

Adjusting Default Partition Parameters

This is supplementary information. You can create and use HSM partitions, using default parameters, without ever referring to this page. However, if you wish to adjust and control the size of your partitions, the information on this page might be helpful.

For command syntax, see "partition create" on page 1 in the *LunaSH Command Reference Guide*.

The procedure for creating partitions is described in the *Configuration Guide*.

Use **hsm show** to see:

- Total HSM storage
- Current memory usage
- Current number of partitions
- Maximum number of partitions allowed

Use **partition list** to see:

- All current application partitions
- Total storage allotted to each
- Total used and available storage on each partition

Size of Partitions

The maximum number of partitions depends on the model of SafeNet Luna Network HSM you purchased. Your HSM can be upgraded with additional partition licenses if your desired configuration calls for them. By default, each partition is assigned an equal share of the total HSM memory. For example, if you purchased a SafeNet

Luna Network HSM with 16MB of memory and 10 partition licenses, each partition would have a default size of 1.6 MB. The basic allotment ensures that you can create all licensed partitions, each with enough space to hold at least one RSA key pair.

> **NOTE** Each partition requires approximately 9KB of memory to store security and identity information. Take this into account when creating very small specialized partitions (for example, a partition containing a single key pair for signing and verification).

Creating Custom-Sized Partitions

To specify the amount of memory allotted to a new partition, use **partition create**, including the **-size** option and the desired size in bytes:

lunash:> **partition create -par mypartition -size 100000**

To create a partition that uses all available remaining memory, include the **-allfreestorage** option:

lunash:> **partition create -par mypartition -allfreestorage**

Resizing Partitions

> **CAUTION!** If you intend to resize partitions, be sure to backup the contents of your HSM first. If a partition is at or near capacity, it might be necessary to remove some objects before resizing. You may need to restore the partition from backup after it has been resized.

To specify the amount of memory allotted to an existing partition, use **partition resize**, including the **-size** option and the desired size in bytes:

lunash:> **partition resize -par mypartition -size 50000**

You must specify either the **-size** or **-allfreestorage** option when resizing a partition. You can reduce the size of a partition as long as the desired size is not less than the memory currently in use.

Example with four equal partitions using all storage

If you prefer to have all your partitions sized equally, and to let the HSM do the calculations, the following procedure might be of some value.

To create four equal-size partitions, using all the available storage:

1. Start by creating 20 partitions (the maximum allowed) – each will have X bytes available to it.
2. Delete 4 of them (leaving 16).
3. Resize one partition to use **-allfreestorage**, which makes that partition large (as large as five small partitions - the four partitions you just deleted, freeing their allotment, plus the one you are currently resizing) and leaves the HSM with 15 partitions having X bytes each, plus the large one.
4. Delete another four small partitions.
5. Resize one small partition to use **-allfreestorage**, which makes that partition large (there are now two large partitions) and leaves the HSM with 10 partitions having X bytes each, plus the two large ones.
6. Delete another four small partitions.

7. Resize one small partition to use -**allfreestorage**, which makes that partition large (there are now three large partitions) and leaves the HSM with 5 partitions having X bytes each, plus the three large ones.
8. Delete another four small partitions.
9. Resize the single remaining small partition to use **-allfreestorage**, which makes that partition large and leaves 0 (zero) of the original partitions with X bytes each, and the four large partitions of equal size, and no unallocated space on the HSM.

For the example, we chose conveniently round numbers. You might have a few bytes left over, or one partition slightly larger or smaller than the others, depending on the actual configuration of your HSM.

Separation of HSM Workspaces

Depending on the SafeNet Luna HSM and its configuration, the HSM can have three, or more, logical partitions.

> One for the Security Officer (SO)
> One for the Auditor
> One (or more) for applications and Clients

In rare circumstances, the Security Officer might create and keep cryptographic objects, Normally it is not used for "production" cryptographic operations - the SO space is intended for overall HSM-level administration.

The Auditor partition is used to enable and manage secure audit logging activities, and generally has no other function in the HSM.

Application Partitions

An initialized partition has its own SO. The Partition SO manages what happens inside the partition. The HSM SO creates the partition, and deletes it when necessary, but has no other oversight or control of the partition. This distinction is particularly important in cloud scenarios, but is a significant element in separation of roles for any use of an HSM.

Operation

Crypto operations are normally performed from a logged-in session on the HSM. It is possible to create objects without logging in, so long as the CKA_PRIVATE attribute is set to 0 - that is, public objects. You can also delete any object that has CKA_PRIVATE=0. This is as defined in PKCS#11, and is not a security issue.

The restrictions that you expect come into play for objects that are created with CKA_PRIVATE=1, where only the owner is able to delete (or the SO could delete the entire partition containing the objects).

These distinctions can be demonstrated with **CKDemo** commands 1) Open Session, and 3) Login.

The "Open Session" prompt has three options, to choose the partition that you wish to use:

1. Enter your choice (99 or 'FULL' for full help): 1

   SO[0], normal user[1], or audit user[2]?

   If you select "normal user [1]", when opening a session, you are telling the library that you choose to use the user partition which is owned by the partition User (or is shared by the Crypto-Officer and Crypto-User if the partition User has been separated into those two sub-entities).

The session is started, but you have not yet authenticated, and so cannot perform most operations in the session.

The Login prompt has four options, to perform the needed authentication (log into the session that you started above):

1. Enter your choice (99 or 'FULL' for full help): 3

   Security Officer[0]

   Crypto-Officer [1]

   Crypto-User [2]:

   Audit-User [3]:

2. Enter PIN :

If you have chosen the "normal user [1]" partition, when opening the session, then the valid login authentication options are:

> Crypto-Officer (which is the same as partition User (the black PED key for PED-authenticated HSMs) if the Crypto-Officer/Crypto-User distinction is not in force) or

> Crypto User (which is the limited user when the Crypto-Officer/Crypto-User distinction has been invoked).

If you attempt one of the other two authentications, "Security Officer [0]" or "Audit-User [3]", an error message is returned because those are not applicable to the session type (therefore, the partition type) that you selected earlier.

If certificates are created as private objects (CKA_PRIVATE=1), the Crypto User cannot delete them. Also, the Crypto User cannot create fake private objects with CKA_PRIVATE=1. The Crypto User limitations are focused on restricting access to sensitive and/or private keys and objects.

Key Management Commands

LUNA_CREATE_OBJECT:

LUNA_COPY_OBJECT:

LUNA_DESTROY_OBJECT:

LUNA_MODIFY_OBJECT:

LUNA_DESTROY_MULTIPLE_OBJECTS:

LUNA_GENERATE_KEY:

LUNA_GENERATE_KEY_W_VALUE:

LUNA_GENERATE_KEY_PAIR:

LUNA_WRAP_KEY:

LUNA_UNWRAP_KEY:

LUNA_UNWRAP_KEY_W_VALUE:

LUNA_DERIVE_KEY:

LUNA_DERIVE_KEY_W_VALUE:

LUNA_MODIFY_USAGE_COUNT:

Normal Usage Commands

LUNA_ENCRYPT_INIT:

LUNA_ENCRYPT:

LUNA_ENCRYPT_END:

LUNA_ENCRYPT_SINGLEPART:

LUNA_DECRYPT_INIT:

LUNA_DECRYPT:

LUNA_DECRYPT_END:

LUNA_DECRYPT_RAW_RSA:

LUNA_DECRYPT_SINGLEPART:

LUNA_DIGEST_INIT:

LUNA_DIGEST:

LUNA_DIGEST_KEY:

LUNA_DIGEST_END:

LUNA_SIGN_INIT:

LUNA_SIGN:

LUNA_SIGN_END:

LUNA_SIGN_SINGLEPART:

LUNA_VERIFY_INIT:

LUNA_VERIFY:

LUNA_VERIFY_END:

LUNA_VERIFY_SINGLEPART:

LUNA_GET_OBJECT_SIZE:

LUNA_SEED_RANDOM:

Unauthenticated Commands

LUNA_GET:

LUNA_GET_CONTAINER_LIST:

LUNA_GET_CONTAINER_NAME:

LUNA_LOGIN:

LUNA_OPEN_SESSION:

LUNA_PARTITION_SERNUM_GET:

LUNA_FIND_OBJECTS:

LUNA_GET_RANDOM:

LUNA_OPEN_ACCESS:

LUNA_GET_MECH_LIST:

LUNA_GET_MECH_INFO:

LUNA_SELF_TEST:

LUNA_GET_HSM_CAPABILITY_SET:

LUNA_GET_HSM_POLICY_SET:

LUNA_GET_CONTAINER_CAPABILITY_SET:

LUNA_GET_CONTAINER_POLICY_SET:

LUNA_GET_CONFIGURATION_ELEMENT_DESCRIPTION:

LUNA_RETRIEVE_LICENSE_LIST:

LUNA_QUERY_LICENSE:

LUNA_GET_CONTAINER_STATUS:

LUNA_GET_OUID:

LUNA_GET_CONTAINER_STORAGE_INFO:

LUNA_GET_ATTRIBUTE_VALUE:

LUNA_GET_ATTRIBUTE_SIZE:

LUNA_GET_HANDLE:

LUNA_INIT_TOKEN:

LUNA_PARTITION_INIT:

LUNA_CLOSE_ACCESS:

LUNA_DEACTIVATE:

LUNA_MTK_GET_STATE:

LUNA_MTK_RESPLIT:

LUNA_MTK_RESTORE:

LUNA_MTK_UNLOCK_CHALLENGE:

LUNA_MTK_UNLOCK_RESPONSE:

LUNA_MTK_ZEROIZE:

LUNA_CLEAN_ACCESS:

LUNA_PED_GET_SET_RAW_DATA:

LUNA_ZEROIZE:

LUNA_FACTORY_RESET:

LUNA_HA_LOGIN:

LUNA_CONFIGURE_SP:

LUNA_LOG_POLL_HOST:

LUNA_LOG_EXTERNAL:

LUNA_ROLE_STATE_GET:

Commands That are Valid Only in a Session, But Require Special Handling

LUNA_LOGOUT:

LUNA_CLOSE_ALL_SESSIONS:

LUNA_CLOSE_SESSION:

LUNA_GET_SESSION_INFO:

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 Network HSM partition.

All that is required for a Client application to begin using a SafeNet Luna Network 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 Network HSM (port 1792).
2. The Client requests a list of available partitions (if not already known).
3. SafeNet Luna Network 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 Network 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 Network 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 Network 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 Network HSM, look to your application setup for the source of the problem. It might require special configuration. If SafeNet Luna Network 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 Network HSM.

PED Authentication Model

If you have a PED-authenticated SafeNet Luna Network 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" below). 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 Network 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 204).

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 326 and "Partition Capabilities and Policies" on page 86 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.

   role login -name Partition SO

2. Enable **partition policy 22: Allow activation**.

   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):

   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.

   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.

   role logout

To activate a role (Crypto Officer):

1. Login as Crypto Officer (or enter any command that requires authentication).

   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.

   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).

   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.

   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.

   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.

   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.

   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).

   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:

   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**.

   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.

   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.

   role login -name po
3. Set **partition policy 23: Allow auto-activation** to **1**.

   partition changepolicy -policy 23 -value 1

   Auto-activation will begin for each affected role (CO or CU) the next time the role is authenticated.

Other Measures

For best reliability and up-time, in conjunction with the auto-activation option, you can also set "sysconf appliance rebootonpanic enable" on page 1.

Security of Your Partition Challenge

For SafeNet Luna Network 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 Network 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 Logins" on page 363 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.

This is in contrast to revoking a partition from a Client. When a partition's assignment is revoked using the LunaSH command **client revokepartition**, it still exists, but is no longer available to that Client. The partition and its contents could still be used by other Clients, or reassigned to the original Client. See "client revokepartition" on page 1 in the *LunaSH Command Reference Guide* for correct syntax.

To remove a partition from the HSM:

1. Login to LunaSH as admin and list the existing HSM partitions.

   partition list

```
lunash:>partition list

                                                          Storage (bytes)
                                                    ----------------------------
 Partition             Name                      Objects   Total     Used     Free
 ===========================================================================
 154438865287          myLunapar                       0   325896       0   325896
 154438865290          myLunapar2                      0   325896       0   325896


Command Result : 0 (Success)
```

2. Login to the HSM as HSM SO.

   hsm login

3. Delete the partition by specifying its label.

   partition delete -partition <label>

```
lunash:>partition delete -partition myLunapar2
```

```
CAUTION:  Are you sure you wish to delete the partition named:
            myLunapar2
          Type 'proceed' to delete the partition, or 'quit'
          to quit now.
          > proceed
'partition delete' successful.


Command Result : 0 (Success)
```

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 represent the first part of the SafeNet Luna Network HSM with Trusted Path Authentication's two-part, FIPS 140-2 level 3-compliant Client authentication. 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 Network HSM release 7.x requires Luna PED software version 2.7.1 or higher. This software is backward-compatible with legacy SafeNet Luna Network HSM 6.x.

This chapter contains the following sections about PED authentication:

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 200).
- **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 198.
  - **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 199.
  - **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" on page 199.

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

	Password-authentication	PED-authentication
Dual Control	> Not available	> MofN (split-knowledge secret sharing) requires "M" different holders of portions of the role secret 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
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 224.

> **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 roles described in "HSM Roles and Secrets" on page 1 in the *Configuration Guide*. 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 activities and security policies for the HSM: > HSM initialization > Partition creation and assignment to clients > HSM policy management **Mandatory**
	Red SafeNet HSM Domain	HSM Domain or Key Cloning Vector	Cryptographically defines the set of HSMs that can participate in cloning for backup. See "Domain PED Keys" on page 199. **Mandatory**
	Orange SafeNet HSM Remote PED	Remote PED Vector	Establishes a connection to a Remote PED server. **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: > Partition initialization > Crypto Officer/Crypto User role setting > Partition policy management **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 SafeNet HSM Domain	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 page 199. **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 is a read-only role that can perform cryptographic services using keys already existing within the partition. **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**
HSM Auditing	**White** 	Audit User (AU) secret	Authenticates the Audit User role, responsible for audit log management. This role has no access to other HSM services. **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" on the next page)
- 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.

For more information about cloning domains, see "Domain Planning" on page 1 in the *Configuration Guide*.

> **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 224). Copies made later are true copies with the same PED PIN, intended as backups for one person (see "Duplicating Existing PED Keys" on page 234). 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 194.

M of N Split Secrets

The Luna PED can split an authentication secret between multiple PED keys (up to 16), and require a minimum number of the split keys to authenticate the role. This provides a customizable layer of security by requiring multiple trusted people to be present for authentication.

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.

In this scenario, the HSM SO authentication secret is split between 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, some of them, or 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 may 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 185). 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 of PED keys) before normal operations can resume.

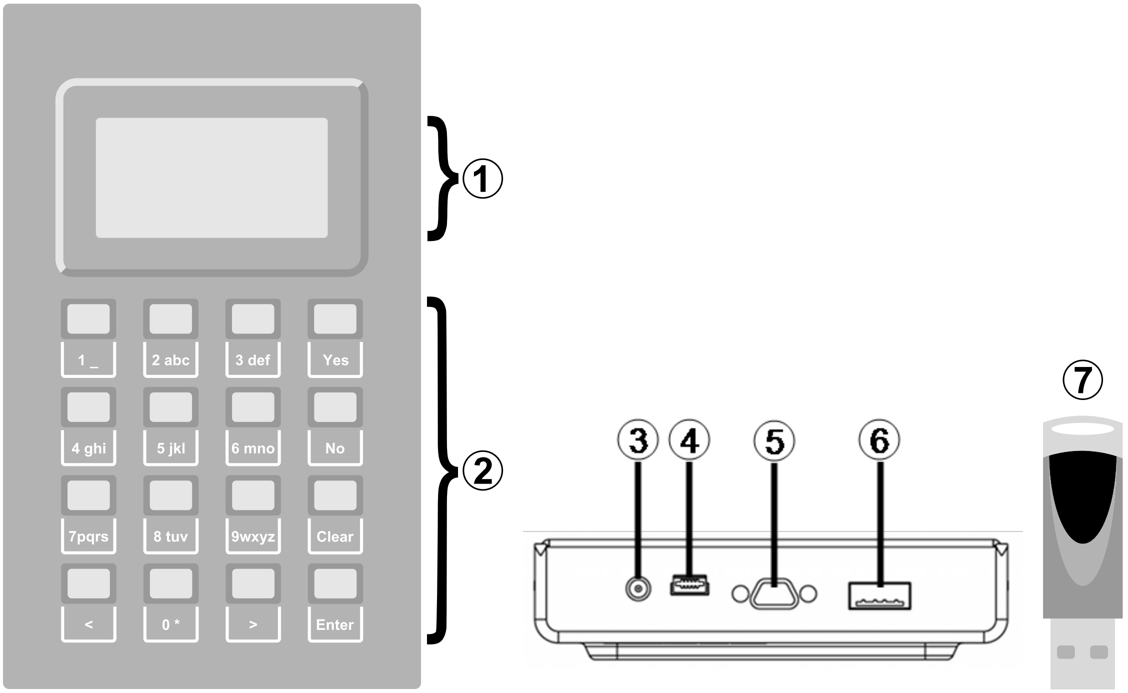
SafeNet Luna PED Hardware Functions

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 202
- "Admin Mode Functions" on page 203

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).

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.

Key	Function
<	> **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 Network 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.

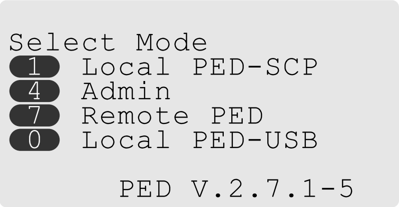
If the Luna PED is connected to an interface when it is powered up, it automatically detects the type of connection being used and switches to the appropriate mode upon receiving the first command from the HSM.

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
```

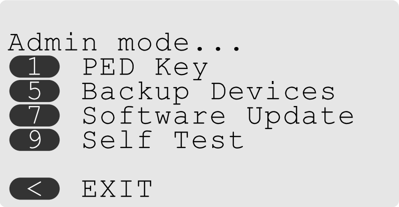
The main menu displays all the available modes, as well as the PED's current firmware version.

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 Network HSM card, or LunaSH/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. Even if you plan to set up Remote PED, you must perform the initial steps in Local PED mode.

Setting Up a Local PED Connection

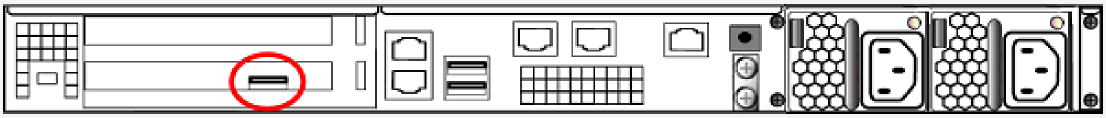
The SafeNet Luna Network HSM administrator can use these directions to set up a Local PED connection. You require:

- 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 appliance.



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 202.

The Luna PED is now ready to perform authentication for the HSM. You may proceed with setting up or deploying your SafeNet Luna Network 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 LunaSH or LunaCM command (see "Performing PED Authentication" on page 229)

> Create copies of your PED keys (see "Duplicating Existing PED Keys" on page 234)

> Change to the Admin Mode to run tests or update PED software (see "Changing Modes" on page 202)

> Prepare to set up a Remote PED server (see "About Remote PED" below)

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 208

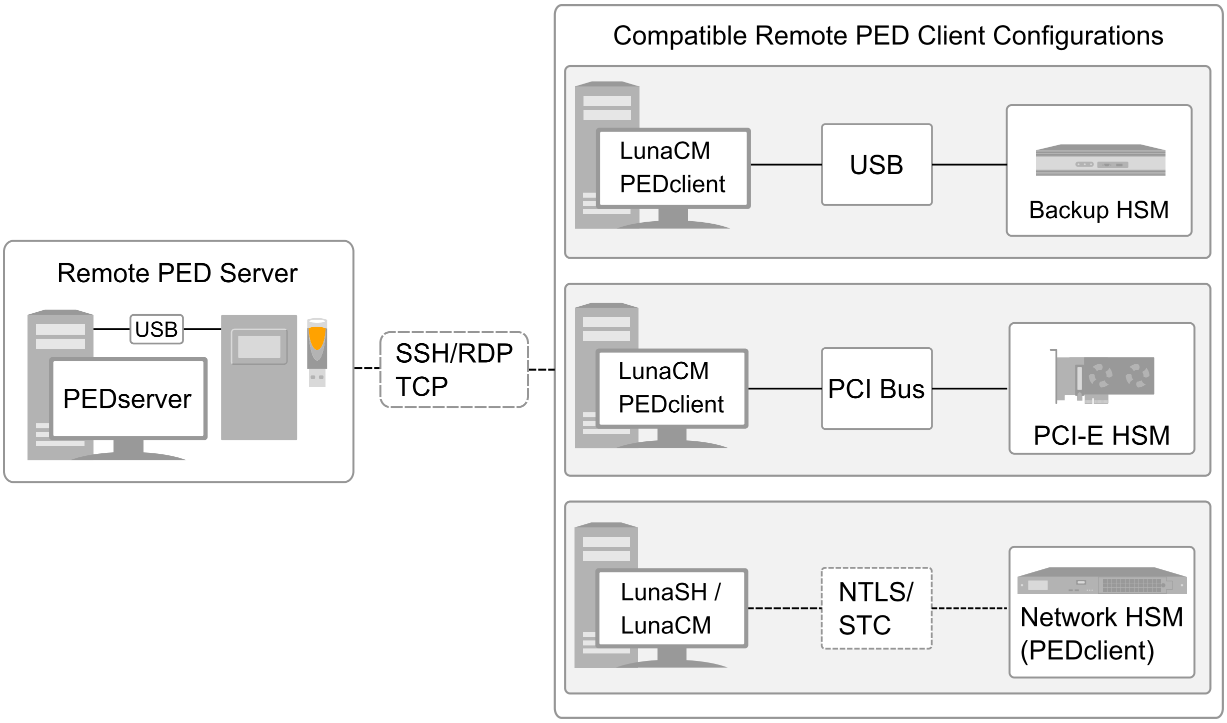
Remote PED Architecture

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).

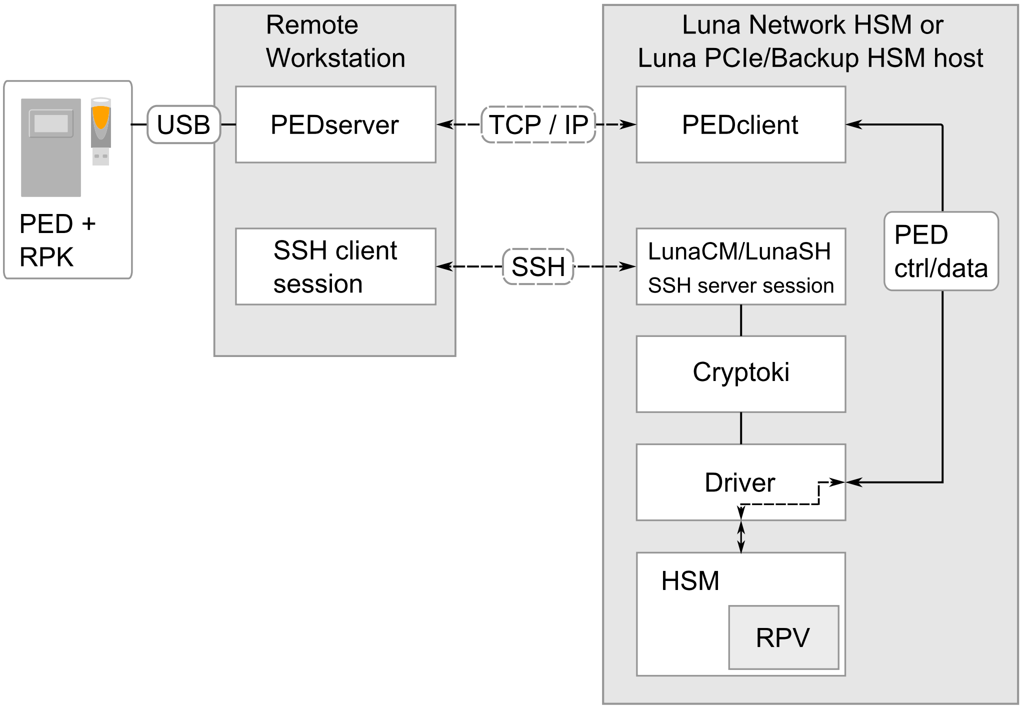
- **Remote PED Key (RPK):** an orange PED key containing an RPV (or multiple PED keys with a split RPV in an M of N implementation).
- **PEDserver:** software that runs on the remote workstation with a USB-connected Luna PED. PEDserver accepts requests from and serves PED actions and data to PEDclient.
- **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 Network HSM 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



Bi-directionality

There are two methods of establishing a Remote PED connection to the HSM:

- **HSM-initiated:** When the HSM requires authentication, it sends (via PEDclient) a request for PED services to the Remote PED host (which receives the request via PEDserver). This requires that the SafeNet Luna Network HSM be allowed to initiate external connections, and that the PEDserver IP port remains open. If the SafeNet Luna Network HSM resides behind a firewall with rules prohibiting these connections, or if your IT policy prohibits opening a port on the Remote PED host, use a PED-initiated connection. See "Opening a Remote PED Connection" on page 212.
- **PED-initiated:** The HSM and Remote PED host exchange and register certificates, creating a trusted connection. This allows the Remote PED host (via PEDserver) to initiate the connection to the SafeNet Luna Network HSM. If you have firewall or other constraints that prevent your HSM from initiating a connection to a Remote PED in the external network, use this connection method. See "Opening a Remote PED Connection" on page 212.

The following constraints apply to PED-initiated connections:

- A maximum of 20 Remote PED servers can be registered in PEDclient.
- A maximum of 80 Network HSM appliances can be registered in PEDserver.
- If the connection is terminated abnormally (for example, a router switch died), there is no auto-reconnection. PEDserver automatically restarts and runs in HSM-initiated connection mode.
- When running in PED-initiated connection mode, PEDserver does not listen for new HSM-initiated connections, for security and to simplify usability.

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 219.

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 219.

Timeout

PEDserver and PEDclient both have configurable timeout settings (default: 1800 seconds). See "pedserver mode config" on page 257 or "hsm ped timeout" on page 1. The utilities are not aware of each other's timeout values, so the briefer value determines the actual timeout duration. Timeout does not apply to PED-initiated Remote PED connections.

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 **hsm ped connect** in LunaSH or **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.	$\{\text{R1} \Vert \text{padding}\}_{Ke}$ ->	
	<- $\{\text{R2} \Vert \text{R1}\}_{Ke}$	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.	$\{\text{padding} \Vert \text{R2}\}_{Ke}$ ->	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.

PEDServer Configuration File

PED-initiated Remote PED introduces a pedServer.ini file. The **Appliances** section manages registered appliances.

> **CAUTION!** Do not edit the pedServer.ini file. If you have any issues, contact Thales Group Technical Support.

```
[Appliances]
ServerCAFile=C:\Program Files\SafeNet\LunaClient\cert\PedServerCAFile.pem
SSLConfigFile=C:\Program Files\SafeNet\LunaClient\openssl.cnf
ServerName00=myHSM
```

```
ServerIP00=192.20.11.78
ServerPort00=9697
CommonCertName00=66331
[RemotePed]
AdminPort=1502
BGProcessShutdownTimeoutSeconds=25
BGProcessStartupTimeoutSeconds=10
ExternalAdminIF=0
ExternalServerIF=1
IdleConnectionTimeoutSeconds=1800
InternalShutdownTimeoutSeconds=10
LogFileError=1
LogFileInfo=1
LogFileName=C:\Program Files\SafeNet\LunaClient\remotePedServerLog.log
LogFileTrace=0
LogFileWarning=1
MaxLogFileSize=4194304
PingInterval=1
PongTimeout=5
RpkSerialNumberQueryTimeout=15
ServerPortValue=1503
SocketReadRspTimeoutSeconds=60
SocketReadTimeoutSeconds=60
SocketWriteTimeoutSeconds=15
```

A new entry in the main Crystoki.ini file points to the location of the pedServer.ini file.

```
[Ped Server]
PedConfigFile = /usr/safenet/lunaclient/data/ped/config
```

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 page 211
3. "Opening a Remote PED Connection" on page 212
   - "HSM-Initiated Remote PED" on page 213
   - "PED-Initiated Remote PED" on page 217
4. [OPTIONAL] "Ending or Switching the Remote PED Connection" on page 219

If you encounter issues with Remote PED, see "Remote PED Troubleshooting" on page 221.

Once Remote PED is set up, see "PED Key Management" on page 224.

Initializing the Remote PED Vector (RPV) and Creating the Orange PED Key

The Remote PED (via PEDserver) authenticates itself to the SafeNet Luna Network HSM with a randomly-generated encrypted value stored on an orange PED key. The SafeNet Luna Network HSM administrator can use these directions to create this key. 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 224 for more information.

> **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.

To initialize the RPV and create the orange PED key:

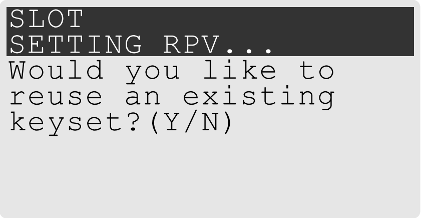
1. If you have not already done so, set up a Local PED connection (see "Local PED Setup" on page 203).
2. Using a serial or SSH connection, log in to the SafeNet Luna Network HSM appliance as **admin**.
3. If the HSM is initialized, login as HSM SO. If not, skip to the next step.

   lunash:>**hsm login**
4. Ensure that you have the orange PED key(s) ready. Initialize the RPV.

   lunash:>**hsm ped vector init**

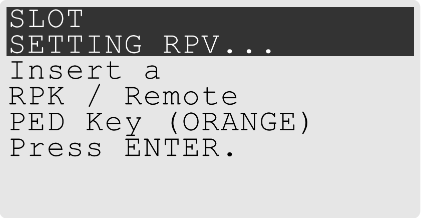
```
lunash:>hsm ped vector init

If you are sure that you wish to initialize remote PED vector (RPV), then enter 'proceed',
otherwise type 'quit'.

> proceed
Proceeding...

Luna PED operation required to initialize remote PED key vector - use orange PED key(s).
```

5. Attend to the Luna PED and respond to the on-screen prompts. See "Creating PED Keys" on page 224 for a full description of the key-creation process.



```
SLOT
SETTING RPV...
Would you like to
reuse an existing
keyset?(Y/N)
```

- 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**.



```
SLOT
SETTING RPV...
Insert a
RPK / Remote
PED Key (ORANGE)
Press ENTER.
```

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.

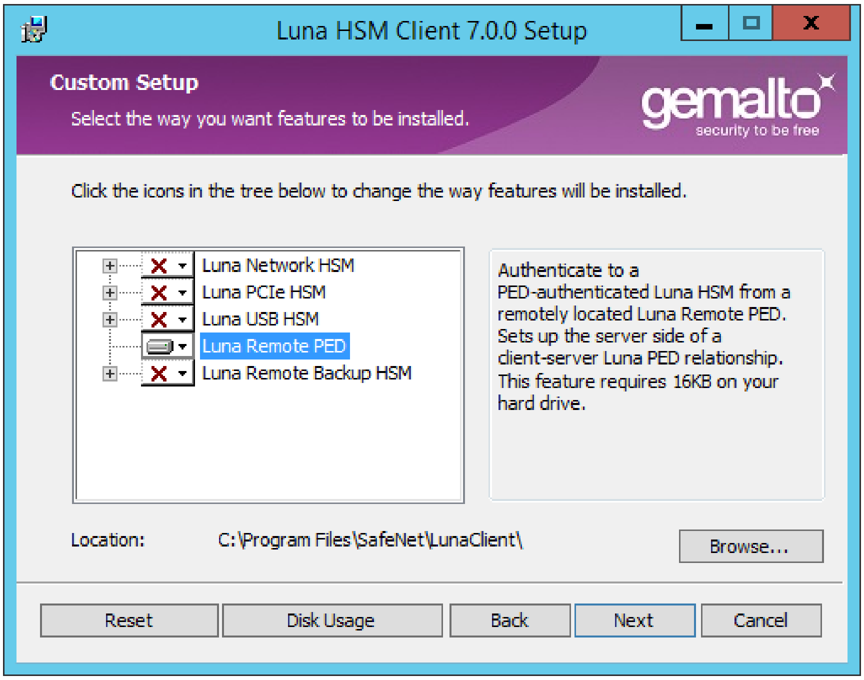
Installing PEDserver and Setting Up the Remote Luna PED

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 7.1 CRN)
- Luna HSM Client installer
- SafeNet Luna PED with firmware 2.7.1 or newer
- USB mini-B to USB-A connector cable
- Luna PED DC power supply

To install PEDserver 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.



Luna HSM Client 7.0.0 Setup

Custom Setup

Select the way you want features to be installed.

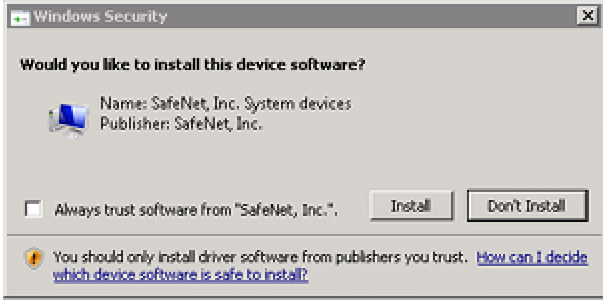
Click the icons in the tree below to change the way features will be installed.

Luna Network HSM
Luna PCIe HSM
Luna USB HSM
Luna Remote PED
Luna Remote Backup HSM

Authenticate to a PED-authenticated Luna HSM from a remotely located Luna Remote PED. Sets up the server side of a client-server Luna PED relationship. This feature requires 16KB on your hard drive.

Location: C:\Program Files\SafeNet\LunaClient\

2. When you are prompted to install the driver, click **Install**.


Windows Security

Would you like to install this device software?

Name: SafeNet, Inc. System devices
Publisher: SafeNet, Inc.

□ Always trust software from "SafeNet, Inc.".

You should only install driver software from publishers you trust. How can I decide which device software is safe to 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. 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 202.

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

There are two methods of establishing a Remote PED connection to the HSM:

> **HSM-initiated:** When the HSM requires authentication, it sends (via PEDclient) a request for PED services to the Remote PED host (which receives the request via PEDserver). This requires that the SafeNet Luna Network HSM be allowed to initiate external connections, and that the PEDserver IP port remains open. If the SafeNet Luna Network HSM resides behind a firewall with rules prohibiting these connections, or if your IT policy prohibits opening a port on the Remote PED host, use a PED-initiated connection.

  See "Opening a Remote PED Connection" above.

> **PED-initiated:** The HSM and Remote PED host exchange and register certificates, creating a trusted connection. This allows the Remote PED host (via PEDserver) to initiate the connection to the SafeNet Luna Network HSM. If you have firewall or other constraints that prevent your HSM from initiating a connection to a Remote PED in the external network, use this connection method.

  See "Opening a Remote PED Connection" above.

If you encounter issues, see "Remote PED Troubleshooting" on page 221.

HSM-Initiated Remote PED

The HSM/client administrator can use this procedure to establish an HSM-initiated Remote PED connection. The procedure is different depending on whether you are setting up Remote PED for the HSM appliance or a client. 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 211)
- Orange PED key with the HSM's RPV (see "Initializing the Remote PED Vector (RPV) and Creating the Orange PED Key" on page 209)
- Administrative access to the SafeNet Luna Network HSM via SSH (if using Remote PED for HSM-level authentication)
- Administrative access to a Luna HSM Client workstation with an assigned user partition (if using Remote PED for partition-level authentication)

To launch PEDserver:

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 251 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 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**

If you are setting up Remote PED with a SafeNet Luna Network HSM appliance, see "To open a Remote PED connection from the SafeNet Luna Network HSM appliance (LunaSH):" below.

If you are setting up Remote PED with a client, see "To open a Remote PED connection from a client workstation (LunaCM):" on the next page.

To open a Remote PED connection from the SafeNet Luna Network HSM appliance (LunaSH):

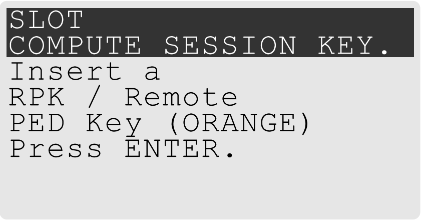
1. Open an SSH session to the SafeNet Luna Network HSM and log in to LunaSH as **admin**.
2. Initiate the Remote PED connection from the SafeNet Luna Network HSM.

   lunash:> **hsm ped connect -ip** <PEDserver_IP> **-port** <PEDserver_port> [**-serial** <serial#>]

   > **NOTE** The **-serial** option is required only if you are using Remote PED to authenticate a SafeNet Luna Backup HSM connected to one of the SafeNet Luna Network HSM's USB ports. If a serial number is not specified, the appliance's internal HSM is used.

   ```
   lunash:>hsm ped connect -ip 192.124.106.100 -port 1503

   Luna PED operation required to connect to Remote PED - use orange PED key(s).
   ```

   The Luna PED prompts for the orange PED key.



```
SLOT
COMPUTE SESSION KEY.
Insert a
RPK / Remote
PED Key (ORANGE)
Press ENTER.
```

3. Present the orange PED key with the correct RPK.

   The HSM authenticates the RPK, and control is returned to the LunaSH prompt.

   ```
   Command Result : 0 (Success)
   ```

   The HSM-initiated Remote PED connection is now open.

4. Verify the Remote PED connection by entering a command that requires PED authentication.

   - If the HSM is already initialized and you have the blue HSM SO key, you can use **hsm login**.
   - If the HSM is uninitialized, you can initialize it now with **hsm init -label** <label>. Have blank or reusable blue and red PED keys ready (or multiple blue and red keys for M of N or to make multiple copies). See "Creating PED Keys" on page 224 for more information on creating PED keys.

   > **NOTE** The HSM-initiated 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 LunaSH to use each time you connect. To drop the Remote PED connection manually, see "Ending or Switching the Remote PED Connection" on page 219.

5. [OPTIONAL] Set a default IP address and/or port for the SafeNet Luna Network HSM to look for a configured Remote PED.

   lunash:>**hsm ped set -ip** <PEDserver_IP> **-port** <PEDserver_port>

   ```
   lunash:>hsm ped set -ip 192.124.106.100 -port 1503

   Command Result : 0 (Success)
   ```

   With this default address set, the HSM administrator can use **hsm ped connect** (without specifying the IP/port) to initiate the Remote PED connection. The orange PED key will be required each time.

   > **NOTE** If you want to use the Remote PED to authenticate a different HSM, you must first drop the current connection. See "Ending or Switching the Remote PED Connection" on page 219.

To open a Remote PED connection from a client workstation (LunaCM):

1. Launch LunaCM on the client.

2. Initiate the Remote PED connection.

   lunacm:>**ped connect -ip** <PEDserver_IP> **-port** <PEDserver_port>

   ```
   lunacm:>ped connect -ip 192.124.106.100 -port 1503
   ```

```
Command Result : No Error
```

3. Issue the first command that requires authentication.

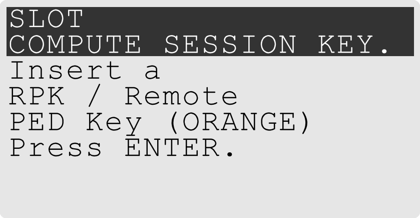
   - If the partition is already initialized and you have the blue Partition SO key, log in.

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

   - If the partition is uninitialized, you can initialize it now. Have blank or reusable blue and red PED keys ready (or multiple blue and red keys for MofN or multiple copies). See "Creating PED Keys" on page 224 for more information on creating PED keys.

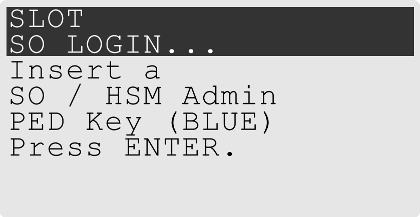
     lunacm:>**partition init -label** <label>

4. 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.
```

5. 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 HSM-initiated 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" on page 219

6. [OPTIONAL] Set a default IP address and/or port for the SafeNet Luna Network HSM to look for a configured Remote PED.

   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** (without specifying the IP/port) to initiate the Remote PED connection. The orange PED key may be required if the RPK has been invalidated on the PED since you last used it.

> **NOTE** If you want to use the Remote PED to authenticate a different HSM, you must first drop the current connection. See "Ending or Switching the Remote PED Connection" on page 219.

PED-Initiated Remote PED

A PED-initiated connection requires the HSM and Remote PED host to exchange and register certificates, creating a trusted connection. This allows the Remote PED host (via PEDserver) to initiate the connection to the SafeNet Luna Network HSM. If you have firewall or other constraints that prevent your HSM from initiating a connection to a Remote PED in the external network, use this connection method. The HSM administrator can use this procedure to set up the 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 211)
- Orange PED key with the HSM's RPV (see "Initializing the Remote PED Vector (RPV) and Creating the Orange PED Key" on page 209)
- Administrative access to the SafeNet Luna Network HSM via SSH

> **NOTE** The PED-initiated Remote PED connection procedure requires **admin** access to the appliance via LunaSH, and therefore this method cannot provide authentication services for client partitions.

To open a PED-initiated Remote PED connection:

1. On the Remote PED host, 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 (**C:\Program Files\SafeNet\LunaClient\**)
3. You will need the Remote PED host's NTLS certificate. If you have already set up an NTLS client connection to the appliance using LunaCM, you can find the certificate in **C:\Program Files\SafeNet\LunaClient\cert\client\** If the certificate is not available, you can generate it with the PEDserver utility.

   > **CAUTION!** If the Remote PED host has registered NTLS partitions on any HSM, regenerating the certificate will cause you to lose contact with your registered NTLS partitions. Use the existing certificate instead.

   >**pedserver -regen -commonname** <name>

   ```
   c:\Program Files\SafeNet\LunaClient>pedserver -regen -commonname RemotePED1
   Ped Server Version 1.0.6 (10006)

   Are you sure you wish to regenerate the client certificate?
   All registered partitions may disappear.

   Are you sure you wish to continue?

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

```
Private Key created and written to: C:\Program
Files\SafeNet\LunaClient\cert\client\RemotePED1Key.pem
Certificate created and written to: C:\Program
Files\SafeNet\LunaClient\cert\client\RemotePED1.pem

Successfully regenerated the client certificate.
```

4. Use **pscp** to securely retrieve the SafeNet Luna Network HSM's NTLS certificate. Enter the appliance's admin account password when prompted. Note the period at the end of the command.

   >**pscp admin@**<appliance_IP>**:server.pem .**

   ```
   c:\Program Files\SafeNet\LunaClient>pscp admin@192.20.11.78:server.pem .
   admin@192.20.11.78's password:

   server.pem                | 1 kB |   1.1 kB/s | ETA: 00:00:00 | 100%
   ```

5. Use **pscp** to securely transfer the Remote PED host's NTLS certificate to the SafeNet Luna Network HSM's **admin** account.

   >**pscp .\cert\client\**<certname> **admin@**<appliance_IP>**:**

   ```
   c:\Program Files\SafeNet\LunaClient>pscp .\cert\client\RemotePED1.pem admin@192.20.11.78:
   admin@192.20.11.78's password:

   RemotePED1.pem              | 1 kB |   1.1 kB/s | ETA: 00:00:00 | 100%
   ```

6. Register the SafeNet Luna Network HSM certificate with PEDserver. Use the mandatory **-name** argument to set a unique name for the appliance. The appliance listens for the SSL connection from PEDserver at the default port **9697**.

   >**pedserver -appliance register -name** <appliance_name> **-certificate** <cert_filename> **-ip** <appliance_IP> **-port** <port>

   ```
   c:\Program Files\SafeNet\LunaClient>pedserver -appliance register -name myLunaHSM -certificate
   server.pem -ip 192.20.11.78 -port 9697
   Ped Server Version 1.0.6 (10006)

   Successfully registered host myLunaHSM.
   ```

7. Open an SSH session to the SafeNet Luna Network HSM and log in to LunaSH as **admin**.

8. Register the PEDserver host certificate.

   lunash:>**hsm ped server register -certificate** <certname>

   ```
   lunash:>hsm ped server register -certificate RemotePED1.pem

   'hsm ped server register' successful.

   Command Result : 0 (Success)
   ```

9. Initiate the connection between PEDserver and the SafeNet Luna Network HSM.

   >**pedserver mode connect -name** <appliance_name>

   ```
   c:\Program Files\SafeNet\LunaClient>pedserver mode connect -name myLunaHSM
   Ped Server Version 1.0.6 (10006)

   Connecting to myLunaHSM. Please wait..
   ```

```
Successfully connected to myLunaHSM.
```

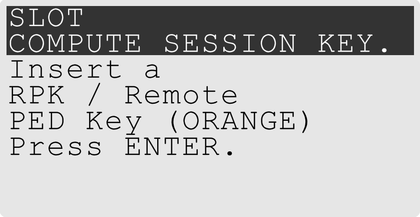
10. Using LunaSH, list the available registered Remote PED servers to find the server name (taken from the certificate filename during registration). Select the server you want to use to authenticate credentials for the appliance.

    lunash:>**hsm ped server list**

    lunash:>**hsm ped select -host** <server_name>

```
lunash:>hsm ped server list

   Number of Registered PED Server :  1

      PED Server  1 : CN = RemotePED1

Command Result : 0 (Success)


lunash:>hsm ped select -host RemotePED1

Luna PED operation required to connect to Remote PED - use orange PED key(s).
```

11. The Luna PED prompts for an orange PED key. Present the orange PED key with the correct RPK for the HSM.



```
SLOT
COMPUTE SESSION KEY.
Insert a
RPK / Remote
PED Key (ORANGE)
Press ENTER.
```

    The secure network connection is now in place between PEDserver and the appliance. You may now perform any actions that require Remote PED authentication. The PED-initiated Remote PED connection does not time out as long as PEDserver is running. If you wish to end the connection in order to connect to a different instance of PEDserver, see "Ending or Switching the Remote PED Connection" below.

Ending or Switching the Remote PED Connection

PEDserver runs on the Remote PED host until explicitly stopped. PEDclient (running on the SafeNet Luna Network HSM) behaves differently depending on the type of Remote PED connection. 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 Remote PED connection using LunaSH:

1. Execute the following LunaSH command:

   lunash:>**hsm ped disconnect**

```
lunash:>hsm ped disconnect

If you are sure that you wish to disconnect, then enter 'proceed', otherwise type 'quit'.
> proceed
```

```
Proceeding...

Command Result : 0 (Success)
```

2. You are now able to initiate a connection to a different Remote PED host running PEDserver. You will need to present the orange PED key.

   lunash:>**hsm 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>.

To end or switch an HSM-initiated connection using LunaCM:

1. Execute the following LunaCM command:

   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. 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>.

To end or switch a PED-initiated Remote PED connection:

1. Execute the following LunaSH command:

   lunash:>**hsm ped deselect -host** <server_name>

```
lunash:>hsm ped deselect -host RemotePED1

Command Result : 0 (Success)
```

2. Check the available list of Remote PED servers.

   lunash:>**hsm ped server list**

   If the Remote PED you want to use is not in the list, see "PED-Initiated Remote PED" on page 217.

3. The new Remote PED server must initiate the connection to the appliance.

   >**pedserver mode connect -name** <appliance_name>

4. In LunaSH, you are now able to select the new Remote PED server from the available list.

lunash:>**hsm ped select -host** <server_name>

Remote PED Troubleshooting

If you encounter problems at any stage of the Remote PED connection process, refer to this section.

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.

Reconnect HSM-initiated Remote PED Before Issuing Commands

As described in the connection procedures, HSM-initiated 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:

```
lunash:>hsm login

Luna PED operation required to login as HSM Administrator - use Security Officer (blue) PED key.

Error:  'hsm login' failed. (300142 : LUNA_RET_PED_UNPLUGGED)

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

To avoid this error, re-initiate the connection before issuing any commands requiring PED authentication:

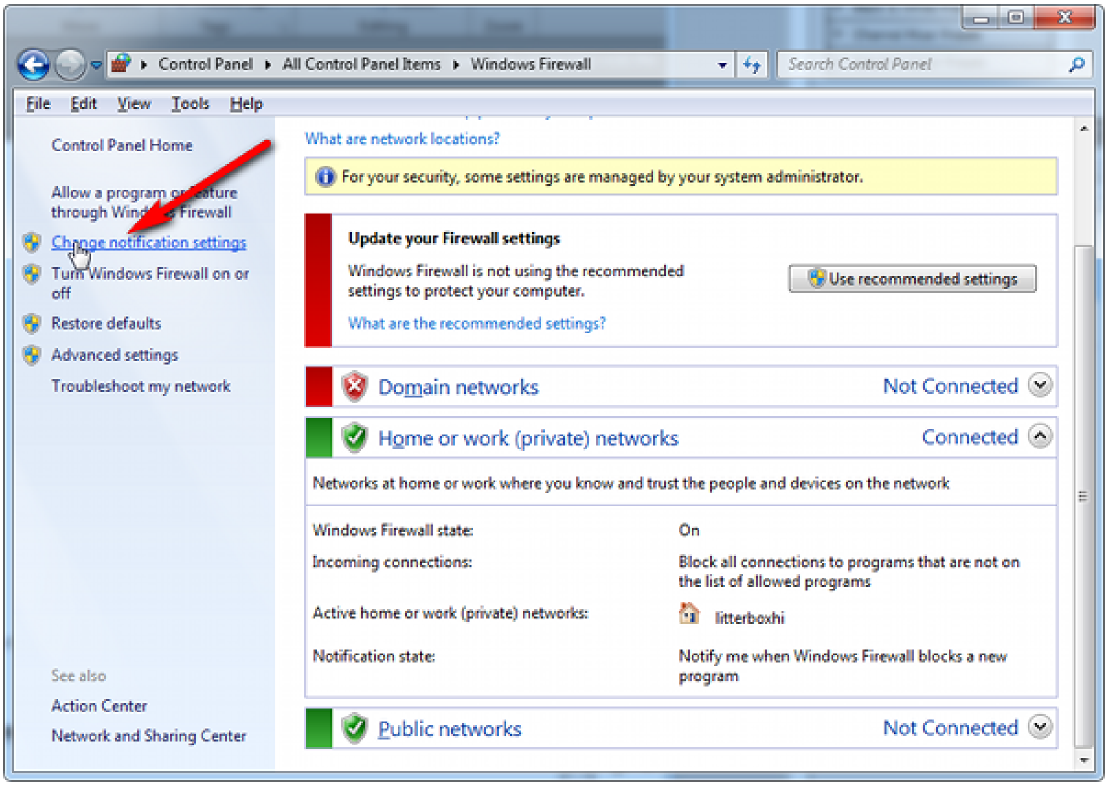
lunash:>**hsm ped connect -ip** <PEDserver_IP> **-port** <PEDserver_port>

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. If your security policy prohibits changes to Windows Firewall, you can use a PED-initiated connection for HSM SO-level operations. See "PED-Initiated Remote PED" on page 217.

1. From the Windows Start Menu, select **Control Panel**.
2. Select **Windows Firewall**.
3. Select **Change notification settings**.


Control Panel ▸ All Control Panel Items ▸ Windows Firewall
Search Control Panel
File Edit View Tools Help
Control Panel Home
Allow a program or feature through Windows Firewall
Change notification settings
Turn Windows Firewall on or off
Restore defaults
Advanced settings
Troubleshoot my network
See also
Action Center
Network and Sharing Center
Use recommended settings

What are network locations?

For your security, some settings are managed by your system administrator.

Update your Firewall settings

Windows Firewall is not using the recommended settings to protect your computer.

What are the recommended settings?

Domain networks — Not Connected

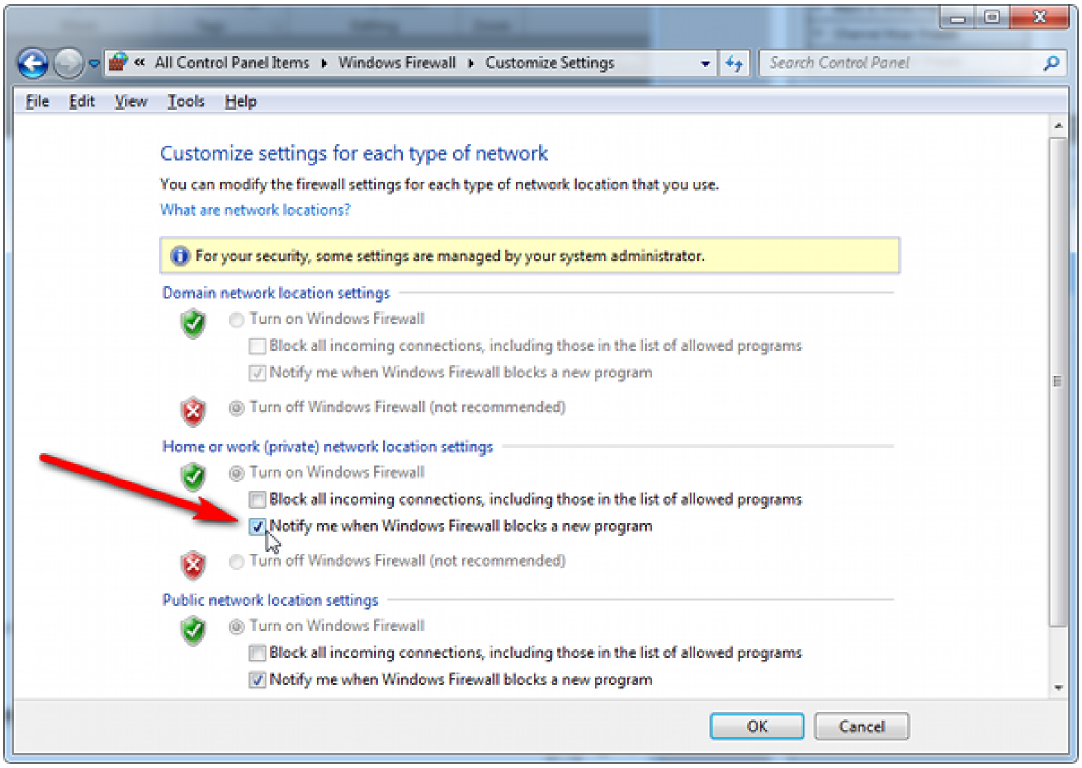
Home or work (private) networks — Connected

Networks at home or work where you know and trust the people and devices on the network

Windows Firewall state:	On
Incoming connections:	Block all connections to programs that are not on the list of allowed programs
Active home or work (private) networks:	litterboxhi
Notification state:	Notify me when Windows Firewall blocks a new program

Public networks — Not Connected

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**.


« All Control Panel Items ▸ Windows Firewall ▸ Customize Settings
Search Control Panel
File Edit View Tools Help
OK
Cancel

Customize settings for each type of network

You can modify the firewall settings for each type of network location that you use.

What are network locations?

For your security, some settings are managed by your system administrator.

Domain network location settings

○ Turn on Windows Firewall

□ Block all incoming connections, including those in the list of allowed programs

☑ Notify me when Windows Firewall blocks a new program

● Turn off Windows Firewall (not recommended)

Home or work (private) network location settings

● Turn on Windows Firewall

□ Block all incoming connections, including those in the list of allowed programs

☑ Notify me when Windows Firewall blocks a new program

○ Turn off Windows Firewall (not recommended)

Public network location settings

● Turn on Windows Firewall

□ Block all incoming connections, including those in the list of allowed programs

☑ 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) or **hsm ped connect** (LunaSH) 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 Network 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. Login to the appliance as **admin** and open the HSM-initiated connection.

   lunash:>**hsm 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 Network 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 -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.

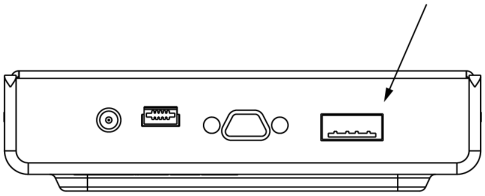
- "Creating PED Keys" below
  - "Stage 1: Reusing Existing PED Keys" on the next page
  - "Stage 2: Defining M of N" on page 227
  - "Stage 3: Setting a PED PIN" on page 227
  - "Stage 4: Duplicating New PED Keys" on page 229
- "Performing PED Authentication" on page 229
- "Consequences of Losing PED Keys" on page 231
- "Identifying a PED Key Secret" on page 233
- "Duplicating Existing PED Keys" on page 234
- "Changing a PED Key Secret" on page 235

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, 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 LunaSH or LunaCM commands to initialize the applicable role, domain, or vector.
   - **Blue HSM SO and Red HSM Domain Keys:**

     lunash:>**hsm init**
   - **Orange Remote PED Key:**

     lunash:>**hsm ped vector init**

     lunacm:>**ped vector init**
   - **Blue Partition SO and Red Partition Domain Keys:**

     lunacm:>**partition init**
   - **Black Crypto Officer Key:**

     lunacm:>**role init -name co**
   - **Gray Crypto User Key:**

     lunacm:>**role init -name cu**
   - **White Audit User Key:**

     lunash:>**audit init**

   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:

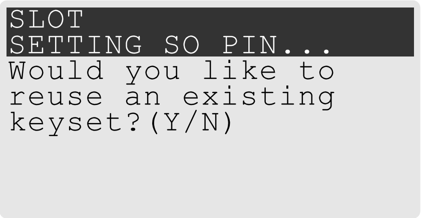
> You want to use the same blue SO key to authenticate multiple HSMs/partitions

> 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.

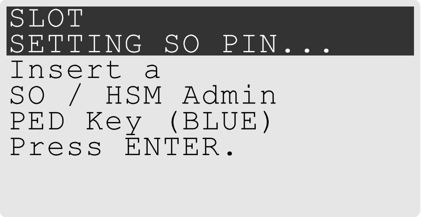
See "Shared PED Key Secrets" on page 198 and "Domain PED Keys" on page 199 for more information.

1. The first PED prompt asks if you want to reuse an existing PED key. Press **Yes** or **No** on the keypad to continue.



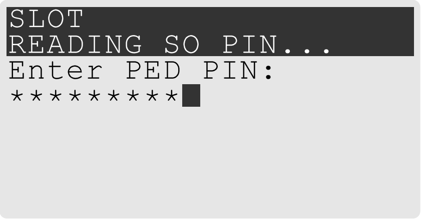
```
SLOT
SETTING SO PIN...
Would you like to
reuse an existing
keyset?(Y/N)
```

- 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**.



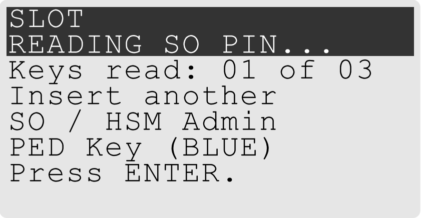
```
SLOT
SETTING SO PIN...
Insert a
SO / HSM Admin
PED Key (BLUE)
Press ENTER.
```

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**.



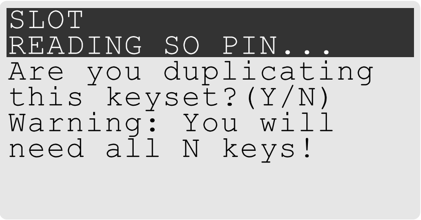
```
SLOT
READING SO PIN...
Enter PED PIN:
*********█
```

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.



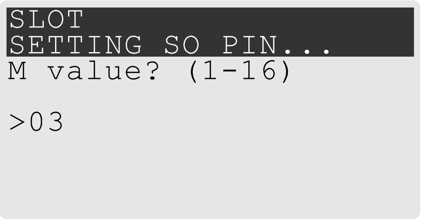
```
SLOT
READING SO PIN...
Are you duplicating
this keyset?(Y/N)
Warning: You will
need all N keys!
```

- If you select **No**, the process is complete.
- If you select **Yes**, complete "Stage 3: Setting a PED PIN" below for all the duplicate keys you want.

Stage 2: Defining M of N

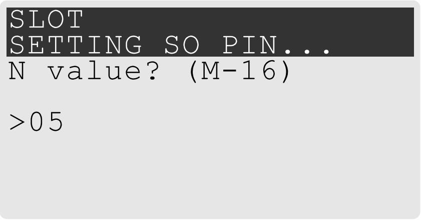
If you chose to create a new keyset, the Luna PED prompts you to define the M of N scheme for the role, domain, or vector. See "M of N Split Secrets" on page 199 for more information. If you do not want to use M of N (authentication by one PED key), enter a value of **1** for both M and N.

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). 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. 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
```

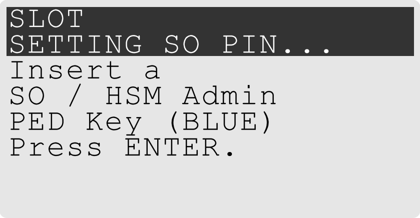
3. Continue to "Stage 3: Setting a PED PIN" below. You must complete stage 3 for each key in the M of N scheme.

Stage 3: Setting a PED PIN

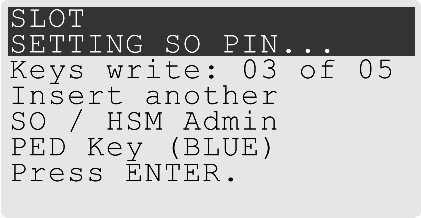
If you are creating a new key or M of N split, you have the option of setting a PED PIN that must be entered by the key owner during authentication. PED PINs must be 4-48 digits long. Do not use 0 for the first digit. See "PED PINs" on page 199 for more information.

> **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 231.

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.

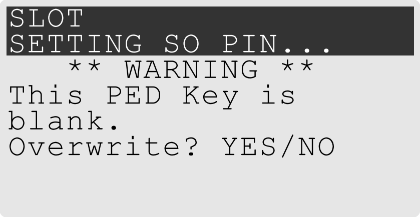


```
SLOT
SETTING SO PIN...
Insert a
SO / HSM Admin
PED Key (BLUE)
Press ENTER.
```



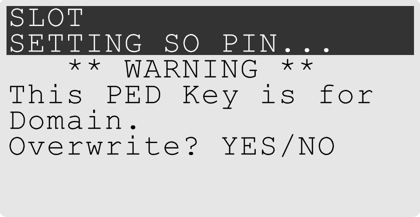
```
SLOT
SETTING SO PIN...
Keys write: 03 of 05
Insert another
SO / HSM Admin
PED Key (BLUE)
Press ENTER.
```

2. Insert the PED key and press **Enter**. The PED prompts for confirmation.

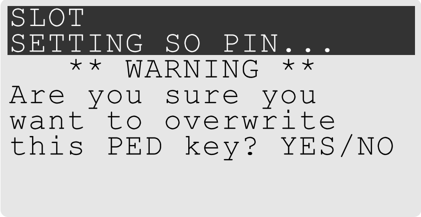


```
SLOT
SETTING SO PIN...
   ** WARNING **
This PED Key is
blank.
Overwrite? YES/NO
```

If the PED key you inserted is not blank, you must confirm twice that you want to overwrite it.



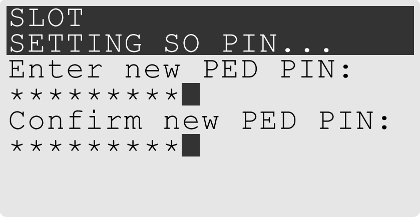
```
SLOT
SETTING SO PIN...
   ** WARNING **
This PED Key is for
Domain.
Overwrite? YES/NO
```



```
SLOT
SETTING SO PIN...
   ** WARNING **
Are you sure you
want to overwrite
this PED key? YES/NO
```

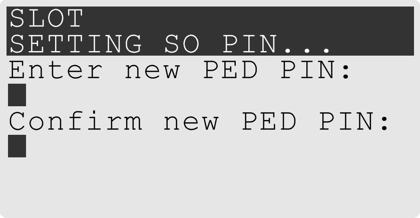
3. The PED prompts you for a PIN.

   - If you want to set a PED PIN, enter it on the keypad and press **Enter**. Enter the PIN again to confirm it.



```
SLOT
SETTING SO PIN...
Enter new PED PIN:
*********█
Confirm new PED PIN:
*********█
```

   - 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.



```
SLOT
SETTING SO PIN...
Enter new PED PIN:
█
Confirm new PED PIN:
█
```

4. If there are more keys in the M of N scheme, repeat this stage. Otherwise, continue to "Stage 4: Duplicating New PED Keys" on the next page.

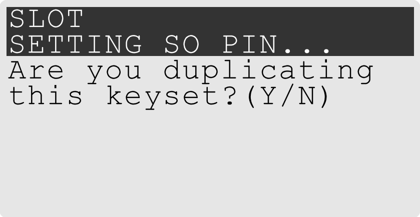
Stage 4: Duplicating New PED Keys

You now have the option to create duplicates of your newly-created PED key(s). There are two reasons to do this now:

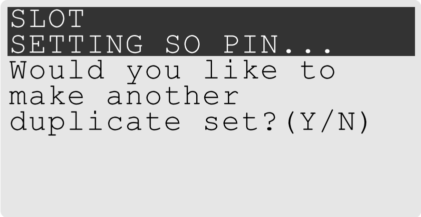
- If you want more than one person to be able to authenticate a role, you can create multiple keys for that role now, with each person being able to set their own PED PIN. Duplicates you create later are intended as backups, and will have the same PED PIN (or none) as the key they are copied from.
- In case of key loss or theft.

You can make backups now or later. See also "Duplicating Existing PED Keys" on page 234.

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.



```
SLOT
SETTING SO PIN...
Are you duplicating
this keyset?(Y/N)
```



```
SLOT
SETTING SO PIN...
Would you like to
make another
duplicate set?(Y/N)
```

   - If you select **No**, the key creation process is complete.
   - If you select **Yes**, complete "Stage 3: Setting a PED PIN" on page 227 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 page 227 for each M of N split. Otherwise, the key creation process is complete.

Performing PED Authentication

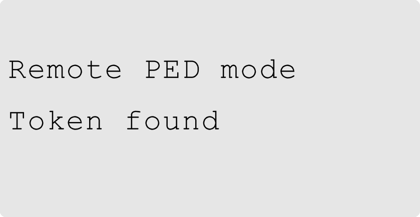
When connected, the SafeNet Luna PED responds to authentication commands in LunaSH or LunaCM. Commands that require PED actions include:

- 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:

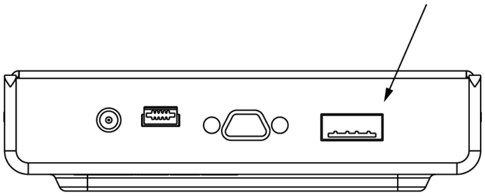
```
lunash:>hsm login

Luna PED operation required to login as HSM Administrator - use Security Officer (blue) PED key.
```

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:

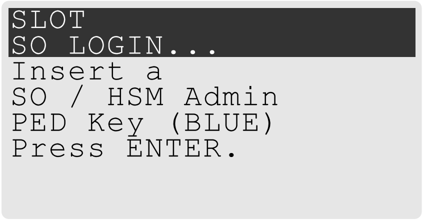


> **CAUTION!** Multiple failed authentication attempts result in zeroization of the HSM or partition, or role lockout, depending on the role. This is a security measure designed to thwart repeated, unauthorized attempts to access cryptographic material. For details, see "Failed Logins" on page 363.

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.
```

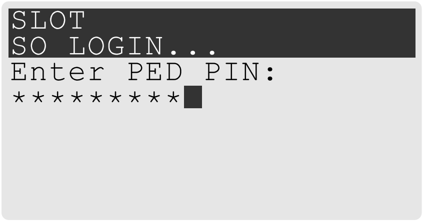


```
SLOT
SO LOGIN...
Insert a
SO / HSM Admin
PED Key (BLUE)
Press ENTER.
```

- 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**.

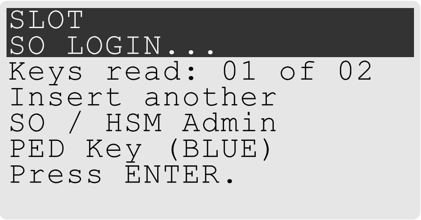


```
SLOT
SO LOGIN...
Enter PED PIN:
*********█
```

- 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**.



```
SLOT
SO LOGIN...
Keys read: 01 of 02
Insert another
SO / HSM Admin
PED Key (BLUE)
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 Network 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" below

> "Red HSM Domain Key" on the next page

> "Orange Remote PED Key" on the next page

> "Blue Partition SO Key" on the next page

> "Red Partition Domain Key" on the next page

> "Black Crypto Officer Key" on page 233

> "Gray Crypto User Key" on page 233

> "White Audit User Key" on page 233

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 SO space 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 SO space 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 SO space(s). If the HSM is factory-reset, the contents of the HSM SO space are unrecoverable. Follow the same procedure as you would if you lost the blue HSM SO key, but you cannot restore the HSM SO space 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 209.

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 may 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 may 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. 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

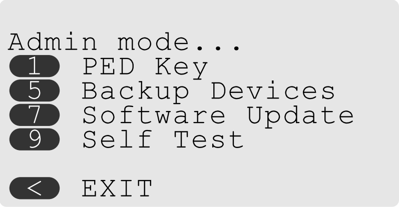
You require:

> SafeNet Luna PED in Admin Mode (see "Changing Modes" on page 202)

> the key you want to identify

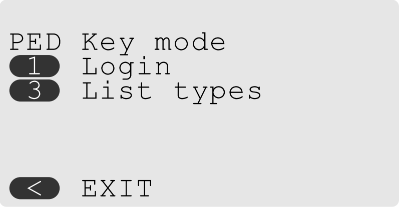
To identify the type of secret stored on a PED key:

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.



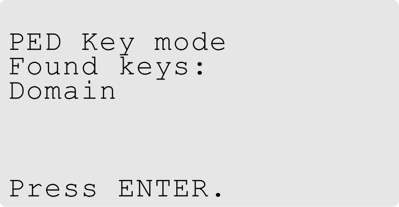
```
Admin mode...
1  PED Key
5  Backup Devices
7  Software Update
9  Self Test

<  EXIT
```

3. From the PED Key mode menu, press **3** on the keypad to select the **List types** option.



```
PED Key mode
1  Login
3  List types



<  EXIT
```

The PED secret type is identified on-screen.



```
PED Key mode
Found keys:
Domain



Press ENTER.
```

Duplicating Existing PED Keys

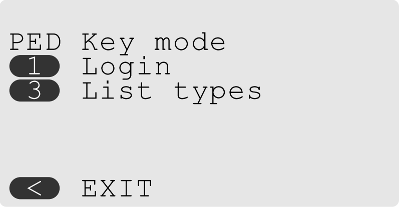
During the key creation process, you have the option to create multiple copies of PED keys. If you want to make backups of your keys later, you can use this procedure to copy PED keys. You require:

- SafeNet Luna PED in Admin Mode (see "Changing Modes" on page 202)
- 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" on page 199.

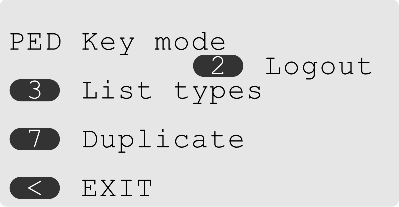
To duplicate an existing PED key:

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" on the next page
- "Orange Remote PED Key" on the next page
- "Blue Partition SO Key" on the next page
- "Red Partition Domain Key" on page 237
- "Black Crypto Officer Key" on page 237
- "Gray Crypto User Key" on page 237
- "White Audit User Key" on page 237

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 LunaSH, log in as HSM SO.

   lunash:>**hsm login**

2. Use the following command to initiate the PED key change:

   lunash:>**hsm changepw**

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 224.

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 SO space 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 LunaSH, log in as HSM SO.

   lunash:>**hsm login**

2. Initialize the RPV.

   lunash:>**hsm 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.

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

2. Use the following command to initiate the PED key change:

   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.

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

2. Use the following command to initiate the PED key change:

   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.

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

2. Use the following command to initiate the PED key change:

   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. Log into LunaSH as **audit**.
2. Log in as the Audit User.

   lunash:>**audit login**

3. Use the following command to initiate the PED key change:

   lunash:>**audit changepwd**

4. 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 "Backup and Restore From the Client to a Remote Backup HSM (LunaCM, RBS)" on page 63 for more information.

See "pedserver" on page 251.

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.

> **NOTE** PEDclient exists on the SafeNet Luna Network HSM appliance, but is not directly exposed. Instead, the relevant features are accessed via LunaSH **hsm ped** commands. See "hsm ped" on page 1 in the *LunaSH Command Reference Guide*.

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 "Backup and Restore From the Client to a Remote Backup HSM (LunaCM, RBS)" on page 63 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 the next page.
config	Modifies or shows existing configuration file settings. See "pedclient mode config" on page 241.
deleteid	Deletes a PED ID mapping. See "pedclient mode deleteid" on page 243.
releaseid	Releases a PED ID mapping from an HSM. See "pedclient mode releaseid" on page 244.
setid	Creates a PED ID mapping. See "pedclient mode setid" on page 245.
show	Queries if a PED Client is currently running and gets details about the PED Client. See "pedclient mode show" on page 246.
start	Starts up the PED Client. See "pedclient mode start" on page 247.
stop	Shuts down an existing PED Client. See "pedclient mode stop" on page 249.
testid	Tests a PED ID mapping. See "pedclient mode testid" on page 250.

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 PED Server.

Syntax

pedclient mode deleteid -id <pedid> [**-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 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 <pedid> [**-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 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 PED Server.

Syntax

pedclient mode setid -id <pedid> **-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 <pedid>	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 a PED Client is currently running and gets details about the PED Client.

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 the PED Client 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 the PED Client.

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 PED Server.

Syntax

pedclient mode testid -id <pedid> [**-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 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 256.
regen	Regenerates the client certificate. Applies to server-initiated (peer-to-peer) mode only. See "pedserver regen" on page 267.

pedserver appliance

Registers or deregisters an appliance, or lists the registered appliances. These commands apply to server-initiated (peer-to-peer) mode only.

Syntax

pedserver appliance

deregister
list
register

Option	Description
deregister	Deregisters an appliance. See "pedserver appliance deregister" on the next page.
list	Lists the registered appliances. See "pedserver appliance list" on page 254.
register	Registers an appliance. See "pedserver appliance register" on page 255

pedserver appliance deregister

Deregister an appliance certificate from PED Server.

Syntax

pedserver appliance deregister -name <unique name> **[-force]**

Option	Description
-name <unique name>	Specifies the name of the appliance to be deregistered from PED Server.
-force	Optional parameter. Suppresses any prompts.

Example

```
C:\Program Files\Safenet\LunaClient>pedServer -appliance register -name hello -force
```

pedserver appliance list

Displays a list of appliances registered with PED Server.

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 PED Server.

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 the 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 259.
disconnect	Disconnects from the appliance. See "pedserver mode disconnect" on page 260.
show	Queries if a PED Server is currently running, and gets details about the PED Server. See "pedserver mode show" on page 261.
start	Starts the PED Server. See "pedserver mode start" on page 263.
stop	Shuts down an existing PED Server. See "pedserver mode stop" on page 265

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 PED Server configuration file.
-set	Updates the PED Server 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 PED Server 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, PED Server certificate) from the PED Server 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 the PED Client 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 PED Server.
-configfile <filename>	Optional. Specifies which PED Server 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 the PED Server 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 PED Server.
-configfile <filename>	Optional. Specifies which PED Server 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 a PED Server is currently running, and gets details about the PED Server.

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 PED Server 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 the PED Server.

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.

Option	Description
-loginfo <0 or 1>	Optional. Specifies if the logger should log "info" 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.
-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>	Specifies the name of the registered appliance to be started with PED Server. Applies to server-initiated (peer-to-peer) mode only.
-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 the PED Server.

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 PED Server will be stopped. Applies to server-initiated (peer-to-peer) mode only.
-configfile <filename>	Optional. Specifies which PED Server 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*) or, if necessary, **vtl createcert** (see "vtl createCert" on page 1 in the *Utilities 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

This chapter describes how to monitor the performance of your HSMs. It contains the following sections:

- "HSM Information Monitor" below

HSM Information Monitor

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 information" on page 1 in the *LunaSH Reference Guide*.

CHAPTER 13: 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
- Installation/application of a destructive Capability Update
- HSM initialization
- 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.

Apply Destructive CUF Update

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 apply a destructive update. All contents 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.

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 rollback and 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 326 .

For command syntax, see "hsm 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 HSMs and Partitions" on page 39 for details.
3. Enter the following command to place the HSM into STM:

   lunash:>**hsm 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.:

   lunash:> **hsm 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 ramdom 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 whether

     -- to proceed with using the HSM
     or, instead,
     -- to perform factory reset and re-initialize the HSM as a safety precaution before proceeding further.

CHAPTER 15:
Secure Trusted Channel (STC)

This chapter describes Secure Trusted Channel (STC). It contains the following sections:

See "Creating an STC Link Between a Client and a Partition" on page 1 in the *Configuration Guide* for detailed procedures that describe how to set up an STC link.

STC Overview

STC protects your HSM/client communications using endpoint and message authentication, verification, and encryption. With STC, HSM/client message integrity is ensured, even when those messages are sent over public or otherwise unsecured networks. You can use STC links to confidently deploy HSM services in cloud environments, or in situations where message integrity is paramount.

When to Use: Comparing NTLS and STC

NTLS and STC connections are best suited for different practical applications. Here are some examples:

NTLS

- Ideally suited for high-performance applications and environments, executing many cryptographic operations per second.
- Best used in traditional data center environments, where the client can be identified by its IP address or hostname.

STC

- Suited for applications with moderate performance requirements

- Preferred where applications are running on physical servers and HSM client credentials are stored on a physical token
- Suited for higher-assurance applications requiring session protection beyond TLS; STC's message integrity and optional additional layer of encryption offers additional protection of client-to-HSM communications
- Best for virtual and cloud environments where virtual machines are frequently cloned, launched, and stopped -- such as when virtual machine auto-scaling is implemented to meet SLAs
- Preferred in "HSM as a Service" environments where multiple customers, departments, or groups all access partitions on a common HSM and want communication to be terminated on the SafeNet Luna HSM card within the appliance

Performance consideration while using STC
STC introduces additional overhead to the communication channel. Depending on the application use case and cryptographic algorithms employed, this could have an impact on application performance.

Security features

STC offers the following security features to ensure the privacy and integrity of your HSM/client communications:

- **Symmetric encryption**. This ensures that only the STC end-points can read data transmitted over an STC link.
- **Message authentication**. Message authentication codes are used to ensure the integrity of the communicated data, to prevent attacks that attempt to add, delete, or modify the messages sent over an STC link.
- **Bi-directional endpoint authentication**. Each endpoint (HSM or client) is assigned a unique identity, which is stored as a hardware or software token. This ensures that only authorized entities can establish an STC connection, and eliminates the risk of a man-in-the-middle attack. See "Client and Partition Identities" on the next page.

Secure tunneling and messaging

STC connections are established in two distinct phases:

1. **Secure tunnel creation**. To ensure client integrity, STC performs bi-directional HSM/client authentication, and creates unique session keys for each STC connection, as described in "Secure Tunnel Creation" on page 282.
2. **Secure message transport**. To ensure message integrity, STC uses symmetric data encryption and message integrity verification, ensuring that any attempt to alter, insert, or drop messages is detected by both end-points, resulting in immediate termination of the connection, as described in "Secure Message Transport" on page 283.

All messages protected outside the HSM

When STC is fully enabled on an HSM, all sensitive communications with the HSM are protected all the way into the HSM. That is, any messages exchanged between a client application and the HSM use STC encryption, authentication, and verification from the client interface to the HSM interface, regardless of whether those links traverse a network, or are internal to an HSM appliance (LunaSH to HSM) or SafeNet Luna HSM client workstation (SafeNet Luna Client to HSM). All STC links that use a network connection also have the same network protection as NTLS links, that is, they are wrapped using SSL.

On a SafeNet Luna Network HSM appliance, there are two separate STC link types, which are configured separately:

- Between the client and a partition. These links are configured as described in "Creating an STC Link Between a Client and a Partition" on page 1 in the *Configuration Guide*. Each client-partition link is configured separately.
- Between the local services and applications running on the appliance (such as LunaSH, NTLS, and the STC service) and the HSM SO partition. This link is called the STC admin channel, and is configured as described in "Establishing and Configuring the STC Admin Channel on a SafeNet Luna Network HSM Appliance" on page 287.

The STC admin channel is local to the appliance, and is used to transmit data between the local services and applications running on the appliance (such as LunaSH, NTLS, and the STC service) and the HSM SO partition.

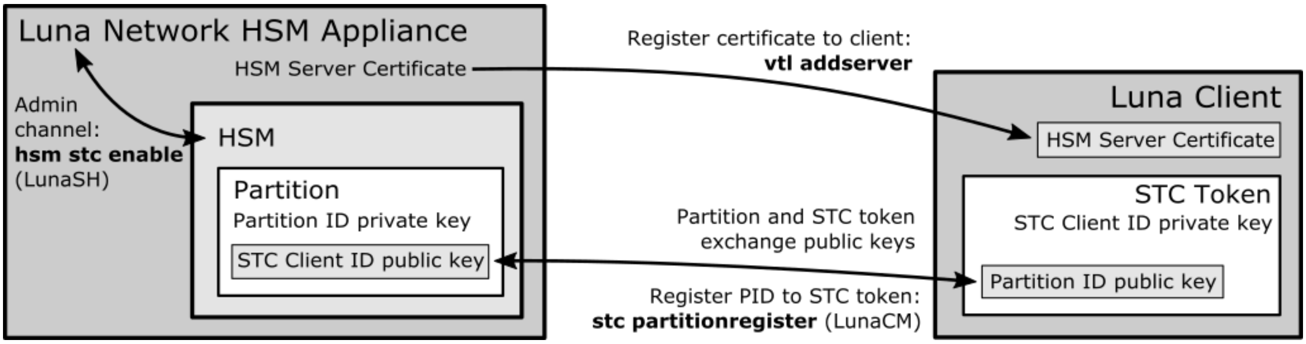
Configurable options

The security features offered by STC are configurable, allowing you to specify the level of security you require, and achieve the correct balance between security and performance. Client/partition STC link parameters are configured using LunaCM. LunaSH/partition STC link parameters are configured using LunaSH.

Client and Partition Identities

The identity of a client or partition at an STC endpoint is defined by a 2048-bit RSA asymmetric public/private key pair, unique to each endpoint. Before you can establish an STC link, you must exchange public keys between the client and partition to establish trust.

Figure 7: Creating an STC Link Between a Client and a Partition


Luna Network HSM Appliance
HSM Server Certificate
Admin channel: **hsm stc enable** (LunaSH)
HSM
Partition
Partition ID private key
STC Client ID public key
Register certificate to client: **vtl addserver**
Partition and STC token exchange public keys
Register PID to STC token: **stc partitionregister** (LunaCM)
Luna Client
HSM Server Certificate
STC Token
STC Client ID private key
Partition ID public key

Partition Identities

The partition private key is always kept in the HSM and is strongly associated with its partition. Only the partition security officer can retrieve the partition's public key for delivery to a client. Upon receipt, the client administrator can use the public key hash to confirm its authenticity, before registering it. You can register multiple partition public keys to a client.

Client Identities

By default, the client's identity pair is stored in a software token on the client's file system, protected by the operating system's access control systems. When using a software token, the client's private key can be moved or copied to another host and used – so any client that possesses this identity pair is considered the authentic client. This enables an elastic client model for many applications.

If you require stronger client authentication, you can choose to use a SafeNet eToken 7300 hardware token to protect the client's private key. When using hard tokens, the client's private key is marked as non-extractable, so only a host with the hard token inserted can successfully authenticate to the HSM partition. The SafeNet eToken 7300 is a FIPS 140-2 Level 3 device.

> **NOTE** After establishing an STC link, the hardware token can be removed from the host computer for safe storage. If the STC link goes down, the hardware token is required to re-establish the link.

Secure Tunnel Creation

Each STC connection is established between a client application and a specific partition on the HSM. As such, each application and partition pair goes through STC tunnel establishment individually. Before STC can create secure tunnels, trust must be established between the client and the partition through the manual exchange of public keys. Once trust has been established, STC links between the client applications and the partition are created.

Establishing Trust Between the Client and the Partition

The trust relationship between the client and the partition is built as follows:

1. When you create a partition, the STC partition identity asymmetric key pair is generated automatically, and stored in the partition.
2. The partition SO extracts the partition's STC public key and provides it (out of band) to the client administrator.
3. The client administrator enables STC on the client machine if not already done.
4. The client administrator registers the partition identity provided in step 2 to the client token (software token or hardware token, as configured). The client administrator can verify the hash of the partition public key before registering it to the client, if desired.
5. The client administrator creates the STC client identity asymmetric key pair, on the client token. This will also automatically export the generated STC client public key to a file.
   - If you are the partition SO, connecting to your un-initialized PSO partition, skip to step 8. Your STC client registration will occur automatically when you initialize the partition.
   - For all others, proceed to step 6.
6. The client administrator takes the client identity public key that was exported automatically during step 5, and provides it (out of band) to the partition SO.
7. The partition SO registers the client's STC identity public key to the partition.
8. The client can now connect to the partition.

> **NOTE** For the partition SO, if this the first time connecting to your uninitialized partition, your client identity will be automatically registered to the partition when you issue the LunaCM **partition initialize** command.

9. Once bi-directional STC public key registration is complete, registered and authorized client applications can establish fully authenticated and confidential STC tunnels with the partition.

Once this sequence is completed the partition will only accept authenticated STC connections from a registered client. You can register additional partitions with this client machine by repeating this process. You can register additional clients to a partition, but any additional client identities need to be registered by the partition SO from a pre-registered client machine.

Recovering lost clients

It is not possible to recover lost clients for PSO partitions as the HSM security officer has no access to the partition once it has been initialized. Therefore, if all registered client tokens to a PSO partition are lost, the only recourse is to have the HSM security officer delete and recreate the partition. The partition objects are lost in this case.

Establishing a Secure Tunnel Between a Client Application and a Partition

Once public keys have been exchanged between a client and a partition, STC is able to establish a secure tunnel between a client application and the partition. To establish a tunnel, the client and partition use secret handshaking to exchange credentials, establish a unique session ID for the tunnel, and create unique message authentication and message encryption keys for the session.

Session Re-Negotiation

Session keys for tunnel are periodically renegotiated, as specified by the STC rekey threshold set for a partition. The rekey threshold specifies the number of API calls, or messages, that can be transmitted over an STC link to the partition before the session keys are renegotiated. You can adjust this value based on your application use cases and security requirements. See "Configuring the Network and Security Settings for an STC Link" on page 294 for more information.

Abnormal Termination

When a client shuts down a connection under normal conditions, it sends a secured message informing the HSM that the connection can be terminated. If a client terminates abnormally, or the network link is lost, the STC Daemon (STCD) detects the abnormal termination, and sends a message to the HSM informing it that the connection has ended, and the connection is closed. If the STCD sends an incorrect connection termination message, the client transparently re-establishes a new STC tunnel.

Secure Message Transport

Once a secure tunnel is established, any messages sent over the STC link are encrypted and authenticated using the unique session keys created when the tunnel is established. In addition, as with NTLS, all STC links use the TLS protocol to secure the link when it traverses a network.

Messages traversing an STC link are protected using Symmetric Encryption and Message Integrity Verification. These features are configurable for each partition and are used for each STC link to that partition. See "Configuring the Network and Security Settings for an STC Link" on page 294 for more information.

Enabling or Disabling STC on the HSM

The STC functionality is enabled or disabled by setting HSM policy 39: Allow Secure Trusted Channel (see "HSM Capabilities and Policies" on page 82). The following instructions are for the HSM SO.

> **NOTE** Enabling **HSM policy 39: Allow Secure Trusted Channel** allows the appliance to use STC or NTLS links between the appliance and its registered partitions. It does not enable STC on the link between the appliance and the HSM (the STC admin channel). If you want to use STC end-to-end (client to HSM) then you must also enable the STC admin channel. See "Establishing and Configuring the STC Admin Channel on a SafeNet Luna Network HSM Appliance" on page 287 for more information.

Enabling STC on the HSM

You can enable STC on the HSM by turning on HSM policy 39: Allow Secure Trusted Channel. Enabling HSM policy 39 allows you to use STC or NTLS to provide the network link between an application partition and a client application. To use STC on a partition, you must also enable STC on the partition by turning on partition policy 37: Force Secure Trusted Channel. See "Enabling or Disabling STC on a Partition" on the next page.

> **NOTE** If you do not plan to use STC in your appliance configuration, do not enable HSM policy 39.

> **NOTE** STC links are not supported over an IPv6 network. You must use NTLS to make partition-client connections via IPv6.

When you enable HSM policy 39: Allow Secure Trusted Channel, the following LunaSH STC commands are blocked, to protect the integrity of any existing STC links:

- **hsm stc identity create**
- **hsm stc identity initialize**
- **hsm stc identity delete**
- **hsm stc identity partition deregister**

> **NOTE** HSM zeroization disables partition policy 39: Allow Secure Trusted Channel. After zeroization, you will need to re-establish your STC links, as described in "Restoring STC After HSM Zeroization" on page 299 and in "Creating an STC Link Between a Client and a Partition" on page 1 in the *Configuration Guide*.

To enable STC on the HSM

1. Login as HSM SO.

   hsm login

2. Turn on HSM policy 39: Allow Secure Trusted Channel, which enables STC on the HSM. Enabling the policy is non-destructive.

   hsm changepolicy -policy 39 -value 1

3. Verify that the policy is enabled:

   hsm showpolicies

   For example:

```
lunash:>hsm showpolicies
.
Description                       Value       Code      Destructive
.
Allow MofN                        On          37        No
Allow Secure Trusted Channel      On          39        No
Allow partition re-initialize     Off         42        No

Command Result : 0 (Success)
```

4. (Optional) Enable the STC admin channel, as described in "Establishing and Configuring the STC Admin Channel on a SafeNet Luna Network HSM Appliance" on page 287.

Disabling STC on the HSM

You can disable STC on the HSM by turning off HSM policy 39: Allow Secure Trusted Channel. Disabling this policy is destructive. It zeroizes the HSM and turns off the ability to use STC to provide the network link between an application partition and a client application, so that only NTLS links are permitted.

To disable STC on the HSM:

1. Login as HSM SO.

   hsm login

2. Turn off HSM policy 39: Allow Secure Trusted Channel, which disables STC on the HSM and zeroizes the HSM.

   hsm changepolicy -policy 39 -value 0

   You are prompted to confirm the action.

3. Verify that the policy is disabled:

   hsm showpolicies

```
lunash:>hsm showpolicies
.
Description                       Value       Code      Destructive
.
Allow MofN                        On          37        No
Allow Secure Trusted Channel      Off         39        No
Allow partition re-initialize     Off         42        No

Command Result : 0 (Success)
```

Enabling or Disabling STC on a Partition

If STC is enabled on the HSM, you can enable STC on the specific partitions on which you want to use STC instead of NTLS. This allows you to use both NTLS and STC links on different partitions on the same HSM. The following instructions are for the Partition SO.

Enabling STC on a Partition

Before you can enable STC on a partition, the HSM SO must enable STC on the HSM, as described in "Enabling or Disabling STC on the HSM" on page 284. The partition SO can then enable STC on a partition by turning on partition policy 37: Force Secure Trusted Channel. Enabling partition policy 37 disables NTLS for the partition and forces it to use STC to provide the network link between the partition and a client application.

To use STC on a partition, you must also create a client token and client identity key pair and exchange and register the partition and client identity public keys between the partition and client, as described in "Secure Trusted Channel (STC) Links" on page 1 in the *Configuration Guide*. Note that the partition token and identity is created automatically when you create a partition, regardless of whether STC is enabled or not.

> **NOTE** HSM zeroization disables partition policy 37: Force Secure Trusted Channel. After zeroization, you will need to re-establish your STC links, as described in "Restoring STC After HSM Zeroization" on page 299 and in "Creating an STC Link Between a Client and a Partition" on page 1 in the *Configuration Guide*.

To enable STC on a partition:

1. Confirm with the HSM SO that STC is enabled on the HSM, as described in "Enabling or Disabling STC on the HSM" on page 284.
2. Run LunaCM, set the active slot to the desired partition, and login as Partition SO.

   slot set slot <slotnum>

   role login -name po
3. Turn on partition policy 37: Force Secure Trusted Channel, which enables STC on the specified partition.

   partition changepolicy -policy 37 -value 1
4. Verify that the policy is enabled:

   partition showpolicies

   ```
   lunash:>partition showpolicies
   .
   Description                                Value       Code
   .
   Allow CBC-PAD (un)wrap keys of any size    On          34
   Force Secure Trusted Channel               On          37

   Command Result : 0 (Success)
   ```

Disabling STC on a Partition

The Partition SO can disable STC on a partition by turning off partition policy 37: Force Secure Trusted Channel. Disabling this policy terminates the existing STC connection to the partition and turns off the ability to use STC to provide the network link between the partition and a client application, so that only NTLS links are permitted.

To disable STC on a partition:

1. In LunaCM, set the active slot to the desired partition, and login as Partition SO:

   slot set slot <slotnum>

role login -name po

2. Turn off HSM policy 37: Allow Secure Trusted Channel, which terminates the existing STC connection to the partition.

   partition changepolicy -policy 37 -value 0

   You are prompted to confirm the action.

3. Verify that the policy is disabled:

   partition showpolicies

```
lunacm:>partition showpolicies
.
Description                                 Value       Code
.
Allow CBC-PAD (un)wrap keys of any size     On          34
Force Secure Trusted Channel                Off         37

Command Result : 0 (Success)
```

Establishing and Configuring the STC Admin Channel on a SafeNet Luna Network HSM Appliance

STC allows you to protect all communications to the HSM, including those that originate on the SafeNet Luna Network HSM appliance, by enabling the STC admin channel. The STC admin channel is local to the appliance, and is used to transmit data between the local services and applications running on the appliance (such as LunaSH, NTLS, and the STC service) and the HSM SO partition. The STC admin channel link is configured separately from the client-partition links, and can be enabled or disabled as required. The following instructions are for the HSM SO.

> **NOTE** Enabling the STC admin channel forces all client-partition links (NTLS or STC) to use STC on the portion of the link from the appliance to the HSM. This may affect NTLS link performance.

Enabling the STC Admin Channel on a SafeNet Luna Network HSM Appliance

When enabled, all communications from the appliance operating system to the HSM are transmitted over the STC admin channel.

> **CAUTION!** Enabling the STC admin channel is service-affecting. It causes an STC service restart, which temporarily terminates all existing STC links to the appliance. It also terminates the existing HSM login session.

To enable the STC admin channel on a SafeNet Luna Network HSM appliance:

1. Open a LunaSH session on the appliance and log in as the HSM SO.

   hsm login

2. Enable the STC admin channel:

   hsm stc enable

```
lunash:>hsm stc enable

    Enabling local STC will require a restart of STC service.
    Any existing STC connections will be terminated.

    Type 'proceed' to enable STC on the admin channel, or 'quit'
    to quit now. > proceed

Successfully enabled STC on the admin channel.

Command Result : 0 (Success)
```

Disabling the STC Admin Channel on a SafeNet Luna Network HSM Appliance

When disabled, all communications from the appliance operating system to the HSM are transmitted, unencrypted, over the local bus.

> **NOTE** Disabling the STC admin channel is service affecting. It causes an STC service restart, which temporarily terminates all existing STC links to the appliance. It also terminates the existing HSM login session.

To disable the STC admin channel on a SafeNet Luna Network HSM appliance:

1. Open a LunaSH session on the appliance and log in as the HSM SO.

   hsm login

2. Disable the STC admin channel:

   hsm stc disable

```
lunash:>hsm stc disable

    Disabling STC on the admin channel will require a restart of STC service.
    Any existing STC connections will be terminated.

    Type 'proceed' to disable STC on the admin channel, or 'quit'
    to quit now. > proceed

Successfully disabled STC on the admin channel.

Command Result : 0 (Success)
```

Configuring the STC Admin Channel on a SafeNet Luna Network HSM Appliance

STC provides several configurable options that define the network settings for an STC link, and the security settings for the messages transmitted over the link. Although default values are provided that provide the optimal balance between security and performance, you can override the defaults, if desired. See "Configuring the Network and Security Settings for an STC Link" on page 294 for more information.

Using a Hard Token to Store the STC Client Identity

By default, STC uses a software token to store the client identity. When using a software token, the client's private key is intentionally portable. That is, it can be moved or copied to another host and used – so any client that possesses this identity pair is considered the authentic client. Allowing this enables an elastic client model

– an important capability for many applications.

Alternatively, you can choose to use a SafeNet eToken 7300 hardware token to protect the client's private key. When using hard tokens, the client's private key is marked as non-extractable, so only a host with the hard token inserted can successfully authenticate to the HSM partition. The SafeNet eToken 7300 is a FIPS 140-2 Level 3 device. The eToken 7300 comes pre-configured for one of two certification types, Common Criteria or FIPS. STC supports the Common Criteria version only.

If you want to use a SafeNet eToken 7300 hardware token to store the client identity, you must initialize the hard token to prepare it for use with STC, as described in "Initializing a SafeNet eToken 7300 Hardware Token" below.

If you want to recover a SafeNet eToken 7300 hardware token that is in a bad state, you must use the SafeNet Authentication Client software to re-initialize the token and reset the default password, as described in "Recovering a SafeNet eToken 7300 Hardware Token" on the next page.

Initializing a SafeNet eToken 7300 Hardware Token

This section describes how to initialize a new (out of the box) SafeNet eToken 7300 for use with STC. Hard token initialization is supported in Windows only. Once the hard token is initialized, you can use it with a Windows, Linux, or Solaris SafeNet Luna Client.

Prerequisites

You require the following software on the workstation used to initialize a SafeNet eToken 7300 hardware token:

> A supported Windows 64-bit operating system

> The SafeNet Luna Client software (6.0 or higher)

> The SafeNet Authentication Client software (64 bit, 8.3 or higher)

To initialize a SafeNet eToken 7300 hardware token:

1. Ensure that the required software is installed on the workstation you are going to use to initialize the token.
2. Edit the **C:\Program Files\SafeNet\LunaClient\crystoki.ini** file to specify the path to the client token library:
   a. Go to the **Secure Trusted Channel** section and add or update the **ClientTokenLib** entry as follows:

      ClientTokenLib=C:\Windows\System32\eToken.dll
3. Insert the SafeNet eToken 7300 token into an available USB slot.
4. Launch LunaCM and enter the following command to verify that the token is recognizable:

   stc tokenlist

   For example:

   - Unitialized token:

     ```
     lunacm:> stc tokenlist

     Token Slot ID  Token Label                                  Serial Number    Initialized
     1                                                           51ea973112       No
     ```

   - Previously initialized token

```
lunacm:> stc tokenlist

Token Slot ID  Token Label                          Serial Number    Initialized
1              stcHWtoken                           51ea973112       Yes
```

5. Enter the following command to initialize the token:

   stc tokeninit -label <label>

   For example:

   - Unitialized token:

     ```
     lunacm:> stc tokeninit stcHWtoken

     Successfully initialized the client token.
     ```

   - Previously initialized token

     ```
     lunacm:> stc tokenlist

     The client token stcHWtoken is already initialized.
     Are you sure you want to re-initialize?
          Type 'proceed' to continue, or 'quit' to quit now --> proceed
     Successfully initialized the client token.
     ```

6. You can now take the token and use it for STC purposes. You can use it in Solaris, Linux, and Windows at this point. You must perform the following tasks on any SafeNet Luna Client workstations on which you intend to use the SafeNet eToken 7300 hardware token:

   a. Install the SafeNet Authentication Client software (8.3 or higher)

   b. Add the following line to the **Secure Trusted Channel** section of the **crystoki.ini** (Windows) or **Chrystoki.conf** (UNIX/Linux) file, to specify the path to the SafeNet Authentication Client eToken library:

Windows	**ClientTokenLib=C:\Windows\System32\eToken.dll**
Linux/UNIX	**ClientTokenLib=<path_to_libeToken.so>** For example, on CentOS, the path is **/usr/lib/libeToken.so**

Recovering a SafeNet eToken 7300 Hardware Token

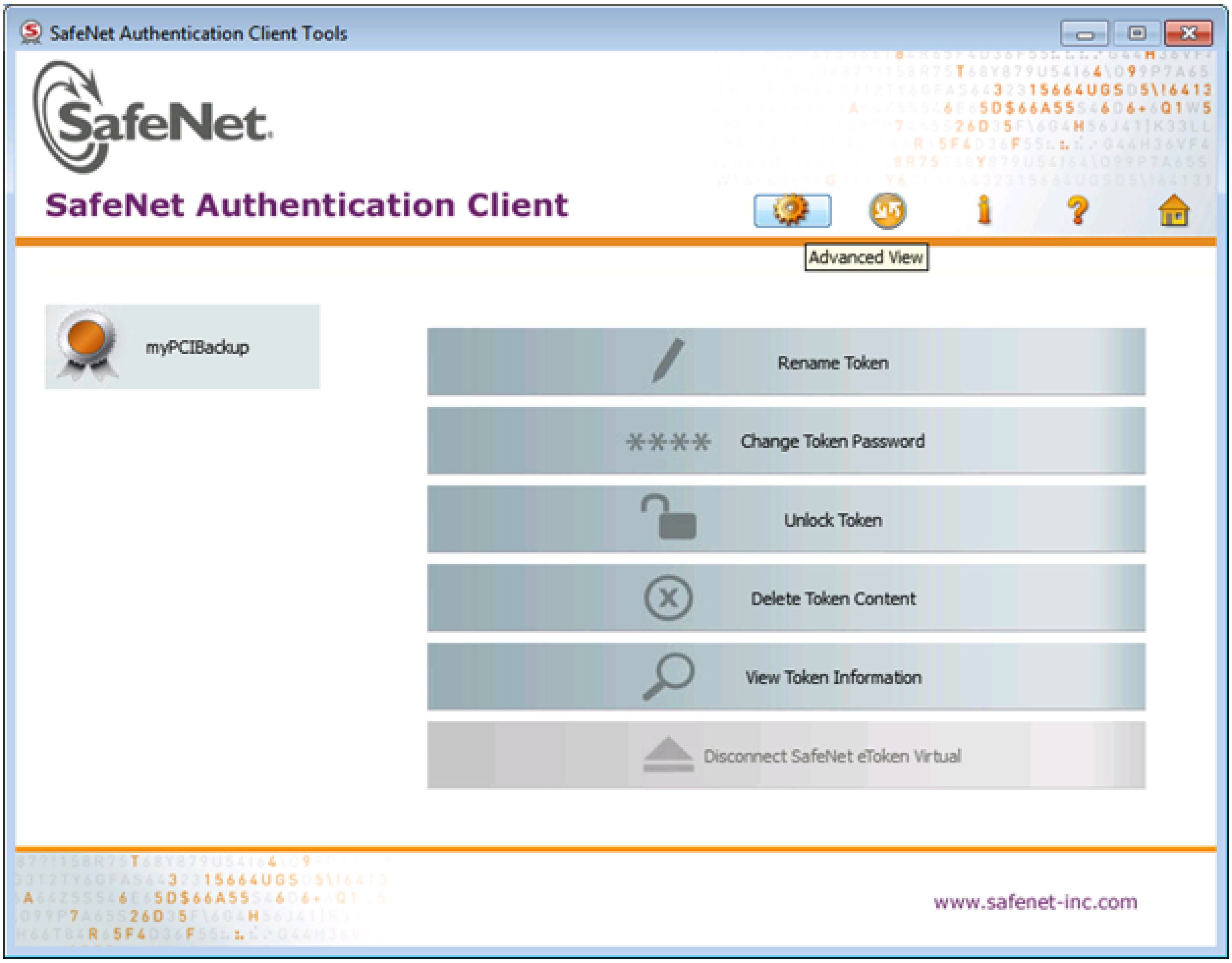
You can use the Windows SafeNet Authentication Client software (8.3 or higher, 64-bit) to recover a SafeNet eToken 7300 that is in an unresponsive state.

To recover an unresponsive SafeNet eToken 7300:

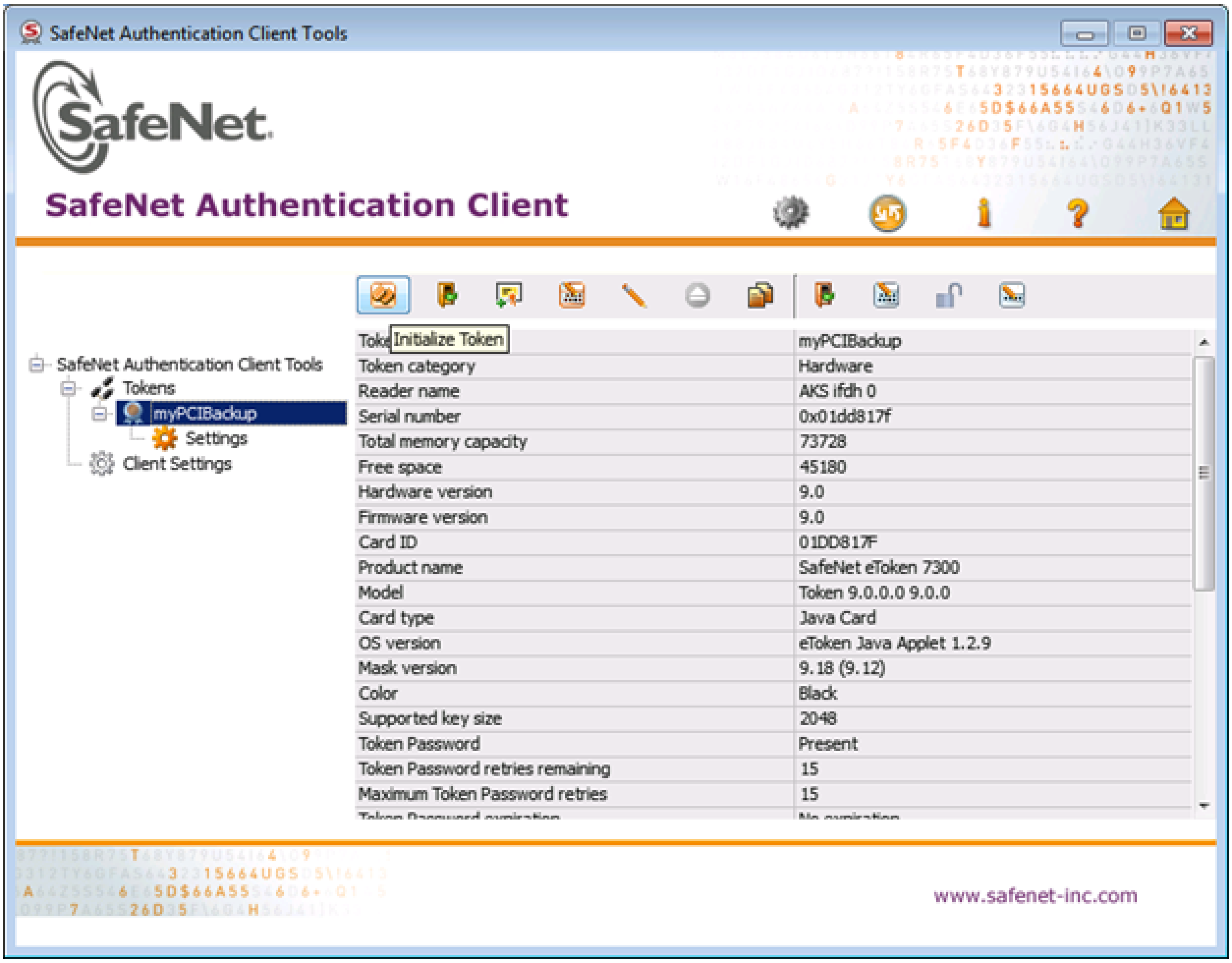
1. Update the registry to add or modify the following entries:

   - HKEY_CURRENT_USER\Software\SAFENET\AUTHENTICATION\SAC\Init\KeepTokenInit = 1
   - HKEY_LOCAL_MACHINE\Software\policies\SAFENET\AUTHENTICATION\SAC\PQ\pqMaxPin = 64
   - HKEY_LOCAL_MACHINE\Software\policies\SAFENET\AUTHENTICATION\SAC\PQ\pqWarnPeriod = 0

2. Launch **SafeNet Authentication Client Tools** from Windows > All Programs > SafeNet > **SafeNet Authentication Client**, and click the **Advanced View** icon.



3. Under the **Tokens** heading in the left-hand column, select the eToken you want to initialize, and click the **Initialize Token** icon to start the initialization.

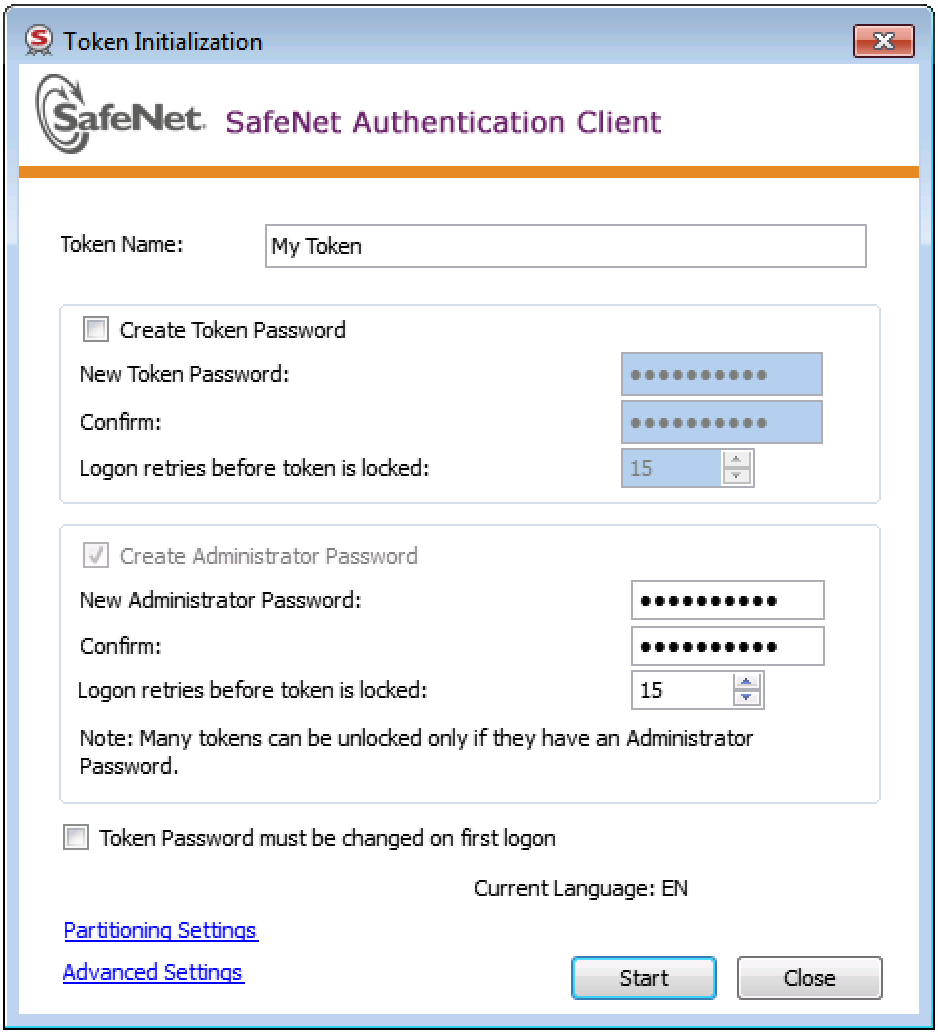

SafeNet Authentication Client Tools
SafeNet
SafeNet Authentication Client
Initialize Token
SafeNet Authentication Client Tools
Tokens
myPCIBackup
Settings
Client Settings
www.safenet-inc.com

Toke	myPCIBackup
Token category	Hardware
Reader name	AKS ifdh 0
Serial number	0x01dd817f
Total memory capacity	73728
Free space	45180
Hardware version	9.0
Firmware version	9.0
Card ID	01DD817F
Product name	SafeNet eToken 7300
Model	Token 9.0.0.0 9.0.0
Card type	Java Card
OS version	eToken Java Applet 1.2.9
Mask version	9.18 (9.12)
Color	Black
Supported key size	2048
Token Password	Present
Token Password retries remaining	15
Maximum Token Password retries	15
Token Password expiration	No expiration

4. On the **Token Initialization** dialog, apply a token name to distinguish this eToken 7300 from other SafeNet STC tokens, and reset the password as follows:

   a. Set the new token password to **password**.

   b. Uncheck the **Token Password must be changed on first logon** checkbox.



Token Initialization

SafeNet Authentication Client

Token Name: My Token

☐ Create Token Password

New Token Password: ●●●●●●●●●●

Confirm: ●●●●●●●●●●

Logon retries before token is locked: 15

☑ Create Administrator Password

New Administrator Password: ●●●●●●●●●●

Confirm: ●●●●●●●●●●

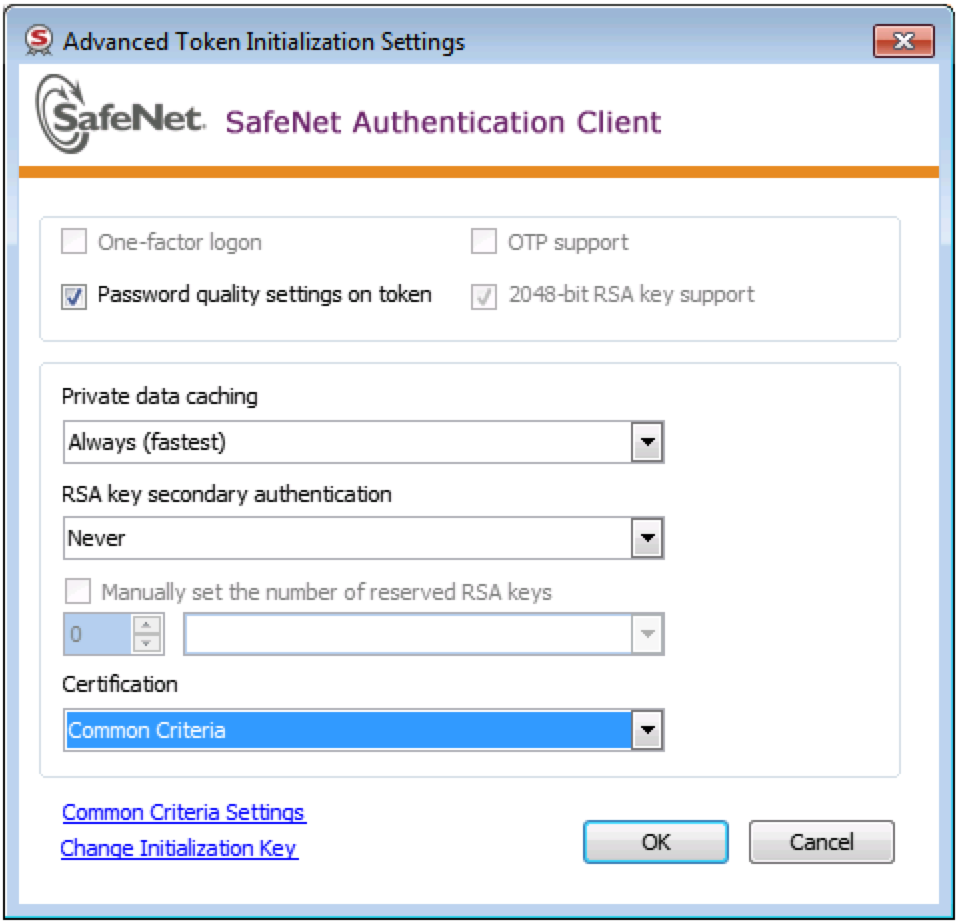
Logon retries before token is locked: 15

Note: Many tokens can be unlocked only if they have an Administrator Password.

☐ Token Password must be changed on first logon

Current Language: EN

5. Select **Advanced Settings** at the bottom left of the dialog.
6. In the **Advanced Settings** dialog, ensure that the **Certification** type matches the type of the eToken (in this case, Common Criteria) and click **OK** to return to the **Token Initialization** dialog.


Advanced Token Initialization Settings
SafeNet
SafeNet Authentication Client
Common Criteria Settings
Change Initialization Key
OK
Cancel

□ One-factor logon
□ OTP support
☑ Password quality settings on token
☑ 2048-bit RSA key support

Private data caching: Always (fastest)

RSA key secondary authentication: Never

□ Manually set the number of reserved RSA keys: 0

Certification: Common Criteria

7. In **Token Initialization**, click **Start** to launch token initialization. Two progress bars are shown followed by a success announcement.

Configuring the Network and Security Settings for an STC Link

STC provides several configurable options that define the network settings for an STC link, and the security settings for the messages transmitted over the link. Although default values are provided that provide the optimal balance between security and performance, you can override the defaults, if desired.

The configurable options are set at the partition level and apply to all STC links to a specific partition. This allows you to configure different settings for individual partitions. You must have SO privileges to the partition to configure its STC options.

For the STC admin channel, the configurable options apply to all communications between the HSM and the local services and applications on the appliance, such as LunaSH and NTLS.

Configurable Options

You can configure the following options for partition/client STC links, or for the STC link between the HSM and the appliance operating system for local services and applications on the appliance, such as LunaSH and NTLS (the STC admin channel).

Use LunaCM to configure the STC options for partitions with SO. Use LunaSH to configure the STC options for partitions owned by the HSM SO, and to configure the link between LunaSH and the HSM.

Link Activation Timeout

The activation timeout is the maximum time allowed to establish the STC link before the channel request is dropped. You can configure this option to specify the activation timeout for all STC links to a partition.

See "stcconfig activationtimeoutset" on page 1 in the *LunaCM Command Reference Guide*.

See the following commands in the *LunaSH Command Reference Guide*:

- "stc activationtimeout set" on page 1 for client-partition links.
- "hsm stc activationtimeout set" on page 1 for the LunaSA admin channel link.

Message Encryption

By default, all messages traversing an STC link are encrypted. You can configure this option to specify the level of encryption used (AES 128, AES 192, or AES 256) on all STC links to a partition, or to disable encryption on all STC links to a partition.

See "stcconfig cipherset" on page 1 in the *LunaCM Command Reference Guide*.

See the following commands in the *LunaSH Command Reference Guide*:

- "stc cipher enable" on page 1 for client-partition links.
- "hsm stc cipher enable" on page 1 for the appliance admin channel link.

Message Integrity Verification

By default, the integrity of all messages traversing an STC link is verified using an HMAC message digest algorithm. You can configure this option to specify the algorithm used (HMAC with SHA 256, or HMAC with SHA 512).

See "stcconfig hmacset" on page 1 in the *LunaCM Command Reference Guide*.

See the following commands in the *LunaSH Command Reference Guide*:

- "stc hmac enable" on page 1 for client-partition links.
- "hsm stc hmac enable" on page 1 for the appliance admin channel link.

Rekey Threshold

The session keys and encryption keys created when an STC tunnel is established are automatically regenerated after the number of messages specified by the rekey threshold have traversed the link. You can configure this option to specify the key life for the session and encryption keys used on all STC links to a partition.

See "stcconfig rekeythresholdset" on page 1 in the *LunaCM Command Reference Guide*.

See the following commands in the *LunaSH Command Reference Guide*:

- "stc rekeythreshold set" on page 1 for client-partition links.
- "hsm stc rekeythreshold set" on page 1 for the appliance admin channel link.

Managing STC Tokens and Identities

Each SafeNet Luna HSM client and partition, (including the HSM SO partition and the SafeNet Luna Network HSM operating system, for the admin channel link) that serves as an STC endpoint has a unique identity, defined by a 2048-bit RSA asymmetric public/private key pair. The STC identity key pair is stored in the STC token associated with the client or partition. Before STC can create secure tunnels, trust must be established between the client and the partition, through the exchange of public keys.

Partition tokens and identities are created automatically.

Client tokens and identities are created manually, using LunaCM. The Client can use either a software token (the default) or a SafeNet eToken 7300 Hardware Token (see "Using a Hard Token to Store the STC Client Identity" on page 288).

Under normal operating conditions, you should not need to recreate the STC tokens or identities. If, however, you want or need to re-create the STC tokens or identities for operational or security reasons, STC provides commands to do so, as follows:

Client Tokens and Identities

Refer to the following commands in the *LunaCM Command Reference Guide*:

Command	Description
stc identitycreate	Create a client identity on the STC client token. See "stc identitycreate" on page 1.
stc identitydelete	Delete a client identity from the STC identity token. See "stc identitydelete" on page 1.
stc identityexport	Export the STC client identify to a file. See "stc identityexport" on page 1.
stc identityshow	Display the client name, public key hash, and registered partitions for the STC client token. See "stc identityshow" on page 1.
stc partitionderegister	Remove a partition identity from the STC client token. See "stc partitionderegister" on page 1.
stc partitionregister	Register a partition to the STC client token. See "stc partitionregister" on page 1
stc tokeninit	Initialize a client token. See "stc tokeninit" on page 1.
stc tokenlist	List the available STC client identity tokens. See "stc tokenlist" on page 1.

STC Admin Channel Identity

Refer to the following commands in the *LunaSH Command Reference Guide*:

Command	Description
hsm stc identity create	Create a STC client identity for the STC admin channel. See "hsm stc identity create" on page 1.
hsm stc identity delete	Delete the STC admin channel client identity. See "hsm stc identity delete" on page 1.
hsm stc identity initialize	Initialize the STC admin channel client token. See "hsm stc identity initialize" on page 1.
hsm stc identity partition deregister	Remove the HSM SO partition identity public key that is currently registered with the STC admin channel client token. See "hsm stc identity partition deregister" on page 1.
hsm stc identity partition register	Register the HSM SO partition identity public key with the STC admin channel client token. See "hsm stc identity partition register" on page 1.
hsm stc identity show	Display the client name, public key hash, and registered partitions for the STC admin channel client token. See "hsm stc identity show" on page 1.

Restoring STC After HSM Zeroization

The HSM partitions contain the registered client identities used to validate STC clients. Since these are not crypto objects, they are not backed up as part of a normal partition backup. When you perform a destructive operation that results in the HSM being zeroized, such as a login failure, application of a destructive capability upgrade (CUF), factory reset, or HSM decommission, the registered client identities are lost. The partitions must be recreated and STC client connections re-established. The Partition SO can then restore the partition objects from backup using the procedure described in "Backup and Restore HSMs and Partitions" on page 39.

If the HSM or partition is zeroized, the following actions occur:

- HSM policy 39: Allow Secure Trusted Channel is turned off.
- If the STC admin channel is enabled, the STC admin partition identity is deleted, breaking the STC link between LunaSH and the HSM SO partition (the admin channel) on the SafeNet Luna Network HSM appliance.
- The STC application partition identities are deleted, breaking the STC links between the application partitions and their registered clients.

See "Creating an STC Link Between a Client and a Partition" on page 1 in the *Configuration Guide* for a detailed description of how to reconfigure your STC links. You do not need to recreate client tokens or identities. Below is a simplified version of the process to reconfigure STC after HSM zeroization:

HSM SO (LunaSH):

1. Login as HSM SO and enable Policy 39: Allow Secure Trusted Channel.

   ```
   hsm changepolicy -policy 39 -value 1
   ```

2. Create the new partition.

   partition create -partition <label>

3. Export the partition identity public key to the file system.

   stc partition export -partition <label>

4. Use **scp** or **pscp** to transfer the partition identity public key (***.pid**) from the appliance and provide it to the client (Partition SO) by secure means.

   > **NOTE** If you restored the appliance's NTLS configuration from backup, you do not need to provide the HSM Server Certificate (**server.pem**). If you choose to regenerate the HSM identity using **sysconf regencert**, you must provide the new certificate to the Partition SO along with the partition identity public key.

5. Optionally, reestablish STC on the HSM admin channel.

   hsm stc enable

   service restart stc

Partition SO:

1. If you received a new HSM Server Certificate (**server.pem**) from the HSM SO, delete the original server identity and register the new one.

   vtl deleteserver -n <server_IP_or_hostname>

   vtl addserver -n <server_IP_or_hostname> **-c** <server_certificate_filename>

2. Run LunaCM and register the new partition identity public key to the STC client identity.

   stc partitionregister -file <partition_identity> [**-label** <partition_label>]

3. Restart LunaCM to see the partition slot.

   clientconfig restart

4. If you registered a new HSM Server Certificate, find the correct server ID and enable STC.

   clientconfig listservers

   stc enable -id <server_ID>

5. Initialize the partition.

   partition init -label <partition_label>

6. Login as Partition SO and register any additional client identity public keys to the partition. You can use the original public key files, unless the client identities have been recreated. If so, the client administrators must provide them.

   stcconfig clientregister -label <client_label> **-file** <client_identity>

7. Provide the partition identity public key (and, if applicable, the HSM Server Certificate) to each additional client administrator by secure means.

Additional Client Administrators:

1. If you received a new HSM Server Certificate (**server.pem**) from the Partition SO, delete the original server identity and register the new one.

   vtl deleteserver -n <server_IP_or_hostname>

   vtl addserver -n <server_IP_or_hostname> **-c** <server_certificate_filename>

2. Run LunaCM and register the new partition ID public key to the STC client identity.

   stc partitionregister -file <partition_identity>

3. If you registered a new HSM Server Certificate, find the correct server ID and enable STC. If not, restart LunaCM to see the partition slot.

   - **clientconfig listservers**

     stc enable -id <server_ID>

   - **clientconfig restart**

Troubleshooting

Restoring STC After HSM Zeroization

The STC client identities are not backed up with the HSM configuration and you must re-register them manually after HSM zeroization. See "Restoring STC After HSM Zeroization" on page 297 for an outline of this process.

Restoring STC After Regenerating the HSM Server Certificate on the SafeNet Luna Network HSM Appliance

If you regenerate the HSM Server Certificate on the appliance (using the command "sysconf regencert" on page 1 in the *LunaSH Command Reference Guide*), you must complete the following steps to restore any STC links to the appliance:

HSM SO:

1. Using LunaSH, restart the NTLS and STC services.

   service restart ntls

   service restart stc

2. Provide the new HSM Server Certificate (**server.pem**) to each client by **scp**, **pscp**, or other secure means.

Clients:

1. Delete the original server identity from the client using the **vtl** utility.

   vtl deleteserver -n <server_IP_or_hostname>

2. Register the new HSM Server Certificate you received from the HSM SO.

   vtl addserver -n <server_IP_or_hostname> **-c** <server_certificate_filename>

3. Run LunaCM, find the new Server ID, and enable STC for the server.

clientconfig listservers

stc enable -id <server_ID>

SAlogin Error

The **salogin** utility is compatible with NTLS-enabled slots only. If you attempt to use the **salogin** utility with an STC-enabled slot, you will get an error similar to the following. See "salogin" on page 1 in the *Utilities Reference Guide* for more information:

```
# ./salogin -o -s 0 -i 1:1 -p userpin
CA_OpenApplicationID: failed to open application id. err 0x80000030
token not present or app id already open?
```

CHAPTER 16:
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 17: Software, Firmware, and Capability Upgrades

This chapter describes how to maintain and update the functionality of your HSMs by performing the following tasks:

- "Software and Firmware Upgrades" below
- "HSM Capability and Partition Upgrades" on page 310

Software and Firmware Upgrades

Your system consists of components that might, from time to time, require updating to newer versions. The newer version might have fixes or functional improvements that are useful or important for your application. The components that might be affected are:

- Client software. See "Client Software Upgrades" below
- SafeNet Luna Network HSM appliance software (the LunaSH command-line shell and its underlying software). See "Appliance Software Upgrades" below.
- SafeNet Luna Network HSM firmware upgrades. See "HSM Firmware Upgrades" on page 306.
- SafeNet Luna Backup HSM firmware. See "Upgrading the SafeNet Luna Backup HSM Firmware" on page 307.

> **CAUTION!** If you require that your SafeNet Luna Network HSM be FIPS-certified, you must use FIPS-certified firmware. Refer to "Customer Release Notes" on page 13 for more information.

Client Software Upgrades

To upgrade 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 (see "SafeNet Luna HSM Client Software Installation" on page 1 in the *Installation Guide*).

The client uninstaller, when invoked on Windows, 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 be able to resume operation without need to manually restore configuration settings and without need to recreate, re-exchange, and re-register client and appliance certificates for NTLS.

Appliance Software Upgrades

Appliance software updates may include an image of the latest HSM firmware, which you may need to install to take advantage of all of the features in a release. If so, when you install the software, a firmware image is also installed onto the appliance file system. This image becomes the default upgrade firmware, and replaces the existing default upgrade firmware stored on the appliance. Note that firmware installation is a separate process (see "HSM Firmware Upgrades" on the next page).

> **NOTE** Appliance software upgrade is a one-way operation. There is currently no way to downgrade the appliance software once a new version is applied. This contrasts with the SafeNet Luna HSM client software, which can be replaced with any version by uninstalling the current version and installing a desired version, and the SafeNet Luna HSM firmware, which can be rolled back to the version that was installed before the currently-installed version.

To upgrade the appliance software:

To update system software and firmware, you must move the updates, in the form of update package files, to SafeNet Luna Network HSM and apply them. Updates are accompanied by instructions that provide detailed update instructions for each component. System and firmware updates require an authentication code, which is provided in a text file accompanying the update package. See "package" on page 1 in the *LunaSH Command Reference Guide* for command syntax.

1. Copy the appliance software package file to the SafeNet Luna Network HSM, as follows:

Linux/UNIX	**scp** <path>/<packagename>**.spkg admin@**<appliance_host_or_IP>**:**
Windows	**pscp** <path>\<packagename>**.spkg admin@**<appliance_host_or_IP>**:**

2. Stop all client applications connected to the SafeNet Luna Network HSM appliance.
3. At the "login as:" prompt, login to the appliance as admin.
4. At the LunaSH prompt, login as HSM SO:

   hsm login
5. [Optional Step] Verify that the file that you copied is present on the SafeNet Luna Network HSM:

   package listfile
6. [Optional Step] Verify the package on the SafeNet Luna Network HSM:

   package verify <filename>**.spkg -authcode** <code_string>

   where <code> is the authorization code from <filename>**.auth**.
7. Install the software upgrade package on the SafeNet Luna Network HSM:

   package update <filename>**.spkg -authcode** <code_string>

   where <code> is the authorization code from <filename>**.auth**.

   The installation/update process requires approximately one and a half minutes. During that time, a series of messages shows the progress of the update.
8. At the end of this process, a message “Software update completed!” appears. If the software update also included a firmware update, then the latest firmware upgrade package is now on the appliance, waiting to be installed in the HSM.
9. Perform a reboot of the SafeNet Luna Network HSM appliance before you upgrade the firmware:

   sysconf appliance reboot
10. If you wish, upgrade the firmware to the newly-installed version:

a. Login to the appliance as admin.

b. At the LunaSH prompt, login as HSM SO:

   hsm login

c. Check that the desired firmware version is ready to install:

   hsm firmware show

d. Upgrade the firmware:

   hsm firmware upgrade

HSM Firmware Upgrades

In general, a new SafeNet Luna Network HSM is delivered with the current FIPS- validated firmware installed on the HSM card, and with the most recent firmware version included - waiting, but not yet installed - on the SafeNet Luna Network HSM hard drive as an optional update. Similarly, when you install a software update package that includes a firmware component, the software is changed and the accompanying new firmware goes into the waiting area on the appliance hard disk, replacing any previous optional firmware.

You can install the firmware image that is waiting on the appliance, or you can download and install a different version, if desired.

If you want to upgrade the firmware on a SafeNet Luna Backup HSM, see "Upgrading the SafeNet Luna Backup HSM Firmware" on the next page.

> **NOTE** It is strongly recommended that your SafeNet Luna Network HSM be powered from an uninterruptible power supply (UPS) when you perform the firmware update. There is a small chance that a power failure during the update command could leave your SafeNet Luna Network HSM in an unrecoverable condition.

To upgrade the HSM firmware:

1. If you are not installing the default upgrade firmware that is waiting on the appliance, obtain the firmware update secure package from Technical Support. Use scp/pscp to upload the package to the SafeNet Luna Network HSM appliance. See "package" on page 1 in the *LunaSH Command Reference Guide* for command syntax.

Linux/UNIX	**scp** <path>/<packagename>**.spkg admin@**<appliance_host_or_IP>**:**
Windows	**pscp** <path>\<packagename>**.spkg admin@**<appliance_host_or_IP>**:**

2. Stop all client applications connected to the SafeNet Luna Network HSM appliance.

3. At the login prompt, log in to the SafeNet Luna Network HSM appliance as admin.

4. Log in as HSM SO:

   hsm login

5. [Optional Step] If you uploaded a new firmware version to the appliance, verify that the file that you copied is present on the SafeNet Luna Network HSM:

   package listfile

6. [Optional Step] If you uploaded a new firmware version to the appliance, verify the package on the SafeNet Luna Network HSM:

   package verify <filename>**.spkg -authcode** <code_string>

   where <code> is the authorization code from <filename>**.auth**.

7. Install the firmware upgrade package on the SafeNet Luna Network HSM.

   > **NOTE** For customers using a service provider model, you can use the **-useevp** option to specify use of OpenSSL EVP (Digital EnVeloPe library) API to validate the update package, rather than invoking the HSM to do so (which would require HSM SO login). See "package update" on page 1 in the *LunaSH Command Reference Guide*.

   package update <flename>**.spkg-authcode** <code_string>

   where <code_string> is the authorization code from <filename>**.auth**.

   The package update process completes in seconds. The firmware package is now on the appliance, waiting to be installed in the HSM.

8. Run the firmware upgrade command:

   hsm firmware upgrade

9. Log in to the HSM:

   hsm login

10. Verify that the change has taken place. The installed firmware should show the desired target version:

    hsm show

Upgrading the SafeNet Luna Backup HSM Firmware

To upgrade the firmware on a SafeNet Luna Backup HSM, use LunaCM on a SafeNet Luna HSM client computer that is connected to the SafeNet Luna HSM and contains a copy of the firmware upgrade (.fuf) file with its associated firmware authentication code (.txt) file.

To upgrade the SafeNet Luna Backup HSM firmware:

1. Copy the firmware file (<fw_filename>.fuf) to the client root directory. Defaults are:
   - Windows: C:\Program Files\SafeNet\LunaClient
   - Linux: /usr/safenet/lunaclient/bin
2. Obtain the firmware authorization code:
   a. Contact Thales Group Technical Support. The firmware authorization code is provided as a text file.
   b. Copy the <fw_authcode_filename>**.txt** file to the client root directory. Defaults are:
   - Windows: C:\Program Files\SafeNet\LunaClient
   - Linux: /usr/safenet/lunaclient/bin
3. Launch LunaCM.
4. If more than one HSM is installed, note which slot is assigned to that HSM and select it.

   slot set -slot <slot_number>

5. Login as HSM SO.

   role login -name so

6. Enter the following command to upgrade the firmware on the HSM:

   hsm updatefw -fuf <fw_filename>**.fuf -authcode** <fw_authcode_filename>**.txt**

Rollback Behavior

When rolling HSM firmware back to an earlier version, the order of the steps you perform is important.

An HSM that receives a firmware update arrives at that condition with any capabilities/features that were part of the HSM before the update was installed. The pre-update record of <firmware version+configuration> is set. If you rollback, you return the HSM to exactly the state that was recorded, prior to the update. All the same capabilities/features would be available, because they were present before the firmware update.

Any capability that you added after a firmware update would be lost, if you then rolled back the firmware, because the pre-update record of <firmware version+configuration> did not include any capability that you added only post-update. In that case:

> If the late-installed capability **is not** dependent on the newer firmware, then you can simply install it again, on the HSM at the rolled-back firmware version, and it will become part of the pre-update record the next time you update firmware.

> If the late-installed capability **is** dependent on the newer firmware, then you must do without that feature/capability until you once more update to a firmware version that can support it. At that time, you will need to re-install that capabilityupgrade.

The following table summarizes the options comparatively.

	Start with this	**If you do this...**	**Result is this**	**If you next do this...**	**Result is this**	**If you next do this...**	**Result is this**	**If you next do this...**	**Result is this**
Example 1 (Read across ==>)	f/w X and Capabilities A, B, C	Update to f/w Y	f/w Y and Capabilities A, B, C	Roll back to f/w X	f/w X and Capabilities A, B, C				
Example 2 (Read across ==>)		Add Capability D (no dependency)	f/w X and Capabilities A, B, C, D	Update to f/w Y	f/w Y and Capabilities A, B, C, D	Roll back to f/w X	f/w X and Capabilities A, B, C, D		
Example 3 (Read across ==>)		Update to f/w Y	f/w Y and Capabilities A, B, C	Add Capability D (no dependency)	f/w Y and Capabilities A, B, C, D	Roll back to f/w X	f/w X and Capabilities A, B, C		
Example 4 (Read across ==>)		Capability E depends on f/w Y; Attempt to add Capability E fails	f/w X and Capabilities A, B, C (unchanged)	Update to f/w Y	f/w Y and Capabilities A, B, C	Add Capability E (depends on f/w Y)	f/w Y and Capabilities A, B, C, E	Roll back to f/w X	f/w X and Capabilities A, B, C

In Example 1, no capabilities change; only the firmware version.

In Example 2, D is added **before** firmware update; therefore the pre-update record includes capability D, so **D survives** firmware update and firmware rollback.

	Start with this	If you do this...	Result is this	If you next do this...	Result is this	If you next do this...	Result is this	If you next do thi s...	Result is this
In Example 3, D is added **after** firmware update, the pre-update record does not include capability D, so **D does not survive** firmware rollback.									
In Example 4, the pre-update record does not include capability E, so E does not survive firmware rollback.									

HSM Capability and Partition Upgrades

SafeNet Luna Network HSMs are shipped from the factory in specific configurations with capabilities to suit your requirements, based on your selections at time of purchase. It can happen that your requirements change over time. You can purchase capability or partition upgrades to enhance your SafeNet Luna Network HSM.

Thales Group provides Capability and Partition upgrades for SafeNet Luna Network HSM through a web-based Entitlement Management System (EMS). Instructions for applying upgrades through this system can be found here:

SafeNet Luna Network HSM Upgrade Guide

You require Admin-level access to the SafeNet Luna Network HSM to apply upgrades.

CHAPTER 18: 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

This section provides an overview of the SNMP implementation and describes how to install the SNMP subagent.

MIB

We provide the following MIBs (management information base):

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 client install dir>** to the MIB directory on your system.

For SafeNet Luna Network HSM, the host is the appliance, so all the above MIBs are in the appliance, to support SNMP.

> **NOTE** Your SNMP application also requires the standard SNMP MIB **SNMPv2-SMI.txt**. Most applications include the MIB. If you do not have it, however, contact your application vendor. It is also freely available for download from the internet.

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:   sa7pw
   Serial #:    66331
   Firmware:    7.1.0

   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


      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


   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
                 7: Allow secret key masking : 0
                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
```

The SafeNet Appliance MIB

The SAFENET-APPLIANCE-MIB defines appliance status information that can be viewed via SNMP. Currently, that consists of the appliance software version number.

The appliance Table

This table provides a list of all the non-HSM host-specific information on the appliance.

Item	Type	Description	Values
appSoftwareVersion	DisplayString	Appliance Software Version number.	-- from factory

For information about the HSM inside the appliance, see"The SafeNet Luna HSM MIB" on page 314.

SNMP Operation and Limitations with SafeNet Luna Network HSM

This page applies only to SafeNet Luna Network HSM which, as a closed system, has its own agent. This contrasts with other SafeNet Luna HSMs that are installed inside a host computer, or USB-connected to a host, and therefore require you to provide an SNMP agent and configure for use with our subagent.

Various LunaSH commands govern the setup and use of SNMP with the SafeNet appliance. You provide your own SNMP application – a standard, open-source tool like net-snmp, or a commercial offering, or one that you develop yourself – and use the commands described below (and on the following pages) to enable and adjust the SNMP agent on-board the SafeNet appliance.

SNMP-Related Commands

Please refer to the LunaSH Appliance Commands in the Reference Section of this Help for syntax and usage descriptions of the following:

- The **sysconf snmp** command has subcommands **enable**, **disable**, **notification**, **show**, **trap**, and **user**.

- The **sysconf snmp notification** command allows viewing and configuring the notifications that can be sent by the SNMP agent. At least one user must be configured before the SNMP agent can be accessed.
- The **sysconf snmp enable** command enables and starts the SNMP service.
- The **sysconf snmp disable** command stops the service.
- The **sysconf snmp show** command shows the current status of the service.
- The **sysconf snmp trap** command has sub-commands to set, show, and clear trap host information.
- The **sysconf snmp user** command allows viewing and configuring the users that can access the SNMP agent. At least one user must be configured before the SNMP agent can be accessed.

> The **service list** command reports a service: "snmpd - SNMP agent service".

> The **service status**, **service stop**, **service start** and **service restart** commands accept the value "snmp" as a **<servicename>** parameter (that is, you can start, stop or restart the snmp service – this represents some overlap with the **sysconf enable** and **disable** commands, but is provided for completeness).

Coverage

The following are some points of interest, with regard to our reporting.

Memory

Swap usage - Covered by UCD-SNMP-MIB under memTotalSwap, memAvailSwap and memMinimumSwap OID

Physical Memory usage - Covered by UCD-SNMP-MIB under memTotalRea, memAvailReal, memTotalFree OID

Errors - Covered by UCD-SNMP-MIB under memSwapError and memSwapErrorMsg OID

Paging

Size of page file - Not covered

Page file usage - Not covered

Paging errors - Not covered
Note: UCD-SNMP-MIB/memory will report all the data that we get from the "free" command.

CPU

% Utilization Threads - Not covered

%user time - Covered by UCD-SNMP-MIB under ssCpuUsr OID

%system time - Covered by UCD-SNMP-MIB under ssCpuSystem OID

Top running processes - Not covered

Network

Interface status - Covered

% utilization - Covered

Bytes in - Not covered

Bytes Out - Not covered

Errors - Covered

Note: All of the above are already covered by the RFC1213-MIB.

Monitoring Internal Hardware failure

We do not currently keep any status on hardware failure.

Environmental

We support only CPU and mother board temperature.

HSM MIB

The above concerns status of various elements of the appliance, outside the contained HSM.

HSM status is separately handled by the SAFENET-HSM-MIB.

In the current implementation, the object ntlsCertExpireNotification has no value. If you query this object, the response is "Snmp No Such Object.

Information about the HSM, retrievable via SNMP, is similar to executing the following commands:

From SafeNet Luna Network HSM (LunaSH) commands:

- **hsm show**
- **hsm showpolicies**
- **hsm displaylicenses**
- **client show**

From the Luna HSM Client (LunaCM) commands:

- **partition showinfo**
- **partition showpolicies**

MIBS You Need for Network Monitoring of SafeNet Luna Network HSM

The following MIBs are not supplied as part of the SafeNet Luna Network HSM build, but can be downloaded from a number of sources. How they are implemented depends on your MIB utility. Support is restricted to active queries (trap captures only reboots).

- LM-SENSORS-MIB
- RFC1213-MIB
- SNMP-FRAMEWORK-MIB
- SNMP-MPD-MIB
- SNMP-TARGET-MIB
- SNMP-USER-BASED-SM-MIB
- SNMPv2-MIB
- SNMP-VIEW-BASED-ACM-MIB

In addition, the SAFENET-APPLIANCE-MIB is included within the SafeNet Luna Network HSM appliance, to report Software Version.

MIBS You Need for Monitoring the Status of the HSM

You require the following MIB to monitor the status of the HSM:

- SAFENET-HSM-MIB.mib

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 19: Tamper Events

SafeNet Luna Network 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 LunaSH 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. The tamper state is indicated by the **HSM Tamper State** field in the output of the LunaSH **hsm show** command. If tamper events have been detected and not cleared, the field will read **Tamper(s) detected**. Use the **hsm tamper show** command to view detailed information for the tamper event, including whether it 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	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 185 and "Partition Capabilities and Policies" on page 86 for more information.

To recover from a tamper

1. Use the following command to display the last tamper event:

   lunash:> **hsm tamper show**

   > **NOTE** The **hsm tamper show** command only shows 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 LunaSH **hsm supportinfo** command.

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:

   lunash:> **hsm tamper clear**
4. If the tamper message indicates that a reset is required, use the LunaSH **sysconf appliance reboot** command to reboot the HSM:

   lunash:> **sysconf appliance reboot**
5. Verify that all tampers have been cleared:

   lunash:> **hsm tamper show**
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 "Create Application Partitions" on page 1.
   b. To re-initialize the partition roles, see "Configure Application Partitions" on page 1.
   c. To restore the partition contents from backup, see "Backup and Restore HSMs and Partitions" on page 39.
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 20: Troubleshooting

This chapter lists the HSM error codes and offers troubleshooting tips for some common issues. It contains the following sections:

General Troubleshooting Tips

Here are just a few quick things to check if you are experiencing problems:

- Ensure that the date and time are set correctly.
- Check that NTLS is bound to the correct Ethernet port. It must be bound to a port if it is to work, and that port must be the one that is connected for NTLS.
- Ensure that the client is registered with the correct ip/hostname (or that you spelled it correctly, didn't accidentally transpose any characters, used only valid characters, etc.).
- Ensure that the client is given access to the correct partition (again, be sure that it is spelled correctly; be careful of similarly named or numbered partitions).
- Ensure that the **sysconf regencert** command was properly executed (with the IP address, if using IP mode).
- Check the output of the syslog for any information on potential problems with **syslog tail**.
- 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.

Error during SSL Connect (RC_OPERATION_TIMED_OUT) logged to /var/log/messages by the SafeNet Luna HSM client

It means that the client did not receive the SSL handshake response from the appliance within 20 seconds (hard coded).

The following is a list of some potential causes:

- Network issue.
- Appliance is under heavy load with connection requests - this can happen at startup/restart, if client applications attempt to (re-)assert hundreds of connections all at once, without staging or staggering them, and the initial setup handshakes take too long for some transactions (start-up bottleneck). After a large number of simultaneous connections has been successfully established, they can be maintained without further problem.
- Appliance is under heavy load servicing crypto requests from connected clients.
- Appliance was powered down (perhaps the power plug was pulled) in the middle of the handshake.
- The client computer might be experiencing high CPU load, causing it to occasionally delay responses to the appliance.

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
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

HSM Error	Hex Code	PKCS#11 or SFNT Defined CKR 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
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

HSM Error	Hex Code	PKCS#11 or SFNT Defined CKR Error
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
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

HSM Error	Hex Code	PKCS#11 or SFNT Defined CKR Error
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
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

HSM Error	Hex Code	PKCS#11 or SFNT Defined CKR 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
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

HSM Error	Hex Code	PKCS#11 or SFNT Defined CKR 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
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

HSM Error	Hex Code	PKCS#11 or SFNT Defined CKR Error
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
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

HSM Error	Hex Code	PKCS#11 or SFNT Defined CKR Error
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
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

HSM Error	Hex Code	PKCS#11 or SFNT Defined CKR 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
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

HSM Error	Hex Code	PKCS#11 or SFNT Defined CKR Error
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
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

HSM Error	Hex Code	PKCS#11 or SFNT Defined CKR Error
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
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

HSM Error	Hex Code	PKCS#11 or SFNT Defined CKR Error
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
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

HSM Error	Hex Code	PKCS#11 or SFNT Defined CKR Error
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
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

HSM Error	Hex Code	PKCS#11 or SFNT Defined CKR Error
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
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

HSM Error	Hex Code	PKCS#11 or SFNT Defined CKR Error
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
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

HSM Error	Hex Code	PKCS#11 or SFNT Defined CKR Error
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
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

HSM Error	Hex Code	PKCS#11 or SFNT Defined CKR Error
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
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

HSM Error	Hex Code	PKCS#11 or SFNT Defined CKR 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

Hex value	Decimal value	Return code/error description
0xC0000107	3221225735	INVALID BUFFER SIZE FOR CHALLENGE
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

Hex value	Decimal value	Return code/error description
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
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

Hex value	Decimal value	Return code/error description
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
0xC0000704	3221227268	SSL LOAD SERVER CERT ERROR
0xC0000705	3221227269	SSL LOAD SERVER PRIVATE KEY ERROR
0xC0000706	3221227270	SSL VALIDATE SERVER PRIVATE KEY ERROR

Hex value	Decimal value	Return code/error description
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

Hex value	Decimal value	Return code/error description
0xc0002000	3221233664	FAILED TO CREATE THREAD
0xc0002001	3221233665	CALLBACK ERROR
0xc0002002	3221233666	UNKNOWN CALLBACK COMMAND
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

Hex value	Decimal value	Return code/error description
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
(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

Code	Name
(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
(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

Code	Name
(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
(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

Code	Name
(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
(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

Code	Name
(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
(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

Code	Name
(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
(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 21:
User and Password Administration

This section describes tasks related to identities in the HSM or HSM partitions, including changing and resetting passwords, events or actions that cause HSM contents to be lost, and so on. It contains the following sections:

- "About Changing HSM and Partition Passwords" below
- "Failed Logins" on the next page
- "Resetting Passwords" on page 365

About Changing HSM and Partition Passwords

From time to time, you might have reason to change the various passwords on the appliance and HSM. This might be because a password has possibly been compromised, lost, or forgotten, or it might be because you have security procedures that mandate password-change intervals.

The two options are:

Action	Description	When used
Resetting PW	A higher authority sets a user's credentials back to a known default value (without requiring the knowledge or cooperation of the affected user).	> Current holder has lost or forgotten his/her credential (forgot a password, misplaced a PED key) > Current credential is known or suspected to have become compromised > Current holder has departed organization
	contrasts with...	
Changing PW	The legitimate holder of the credential is able to log in with current credentials before directing the HSM, under the current logged-in user's own authority, to change that user's credential to a new value.	> Credential holder suspects possible compromise of credential > Credential holder is complying with organization security provisions (such as mandatory password-change interval)

HSM Passwords

Resetting HSM SO Password

There is no provision to reset the HSM Admin password (for Password Authentication) or PED key (for Trusted Path), except to re-initialize the HSM, which zeroizes the contents of the HSM and of all Partitions on that HSM.

Resetting the password/authentication of a role or user requires a higher authority to invoke the reset. On the HSM, there is no authority higher than the SO/HSM Admin.

Changing HSM SO Password

To change the HSM password (for Password Authentication) or the secret on the blue PED key (for Trusted Path), use the **hsm changepw** command. You will be prompted for the current HSM SO credential, so you do not need to log in separately:

```
lunash:> hsm changepw

Luna PED operation required to login as HSM Administrator - use Security Officer (blue) PED key.

Command result : (0) success
```

Partition Passwords

The Partition SO can use the LunaCM command **role resetpw** to reset the Crypto Officer password or black PED key only if HSM policy 15: Enable SO reset of partition PIN is enabled. By default, this policy is not enabled.

Failed Logins and Forgotten Passwords

See "Failed Logins" below.

Appliance

For password changes affecting the appliance, not including the HSM, see "Users and Passwords" on page 1 in the *Appliance Administration Guide*.

Failed Logins

If you fail three consecutive login attempts as HSM Security Officer (or SO), the HSM contents are rendered unrecoverable. This is a security feature meant to thwart repeated, unauthorized attempts to access your cryptographic material. The number is not adjustable.

> **NOTE** The system must actually receive some erroneous/false information before it logs a failed attempt -- if you merely forget to insert a PED key (for PED-authenticated HSMs), or inserted the wrong color key, that is not counted as a failed attempt.

To fail a login attempt on a Password-authenticated HSM, you would need to type an incorrect password. To fail a login attempt on a PED-authenticated HSM, you would need to insert an incorrect PED key of the correct color, type an incorrect PED PIN, or enter an incorrect challenge secret on an activated partition (see "Control the Outcome" on page 365).

As soon as you successfully authenticate, the counter is reset to zero.

> **CAUTION!** SafeNet Luna 7.1's default settings have HSM policy 15: Enable SO reset of partition PIN set to 0. This policy causes the Crypto Officer role to be permanently locked out after too many bad login attempts (default: 10). If this is not the desired outcome, ensure that the HSM SO sets this destructive policy to 1 *before* creating and assigning partitions to clients.

To view a table that compares and contrasts various "deny access" events or actions that are sometimes confused, see "Comparison of Destruction/Denial Actions" on page 124.

Other roles and functions that need authentication on the HSM have their own responses to too many bad authentication attempts. Some functions do not keep a count of bad attempts; the simple failure of a multi-step or time-consuming operation is considered sufficient deterrent to a brute-force attack. The table in the next section summarizes the responses.

HSM Response When You Reach the Bad-attempt Threshold

Role	Threshold (number of tries)	Result of too many bad login attempts	Recovery
HSM SO	3	HSM is zeroized (all HSM objects identities, and all partitions are gone)	HSM must be reinitialized. Contents can be restored from backup(s).
Partition SO	10	Partition is zeroized.	Partition must be reinitialized. Contents can be restored from backup.
Audit	10	Lockout	Unlocked automatically after 10 minutes.
Crypto Officer	10 (can be decreased by Partition SO)	If HSM policy 15: Enable SO reset of partition PIN is set to 1 (enabled), the CO and CU roles are locked out.	CO role must be unlocked and the credential reset by the Partition SO, using **role resetpw -name co**.
		If HSM policy 15: Enable SO reset of partition PIN is set to 0 (disabled), the CO and CU roles are permanently locked out and the partition contents are no longer accessible. This is the default setting.	The partition must be re-initialized, and key material restored from a backup device.
Crypto User	10 (can be decreased by Partition SO)	The CU role is locked out.	CU role must be unlocked and the credential reset by the Crypto Officer, using **role resetpw -name cu**.

Role	Threshold (number of tries)	Result of too many bad login attempts	Recovery
Domain	n/a	Operation fails	Retry the operation with the correct Domain - usually that would be a backup or restore
Remote PED Key	n/a	Operation fails	Retry establishing a Remote PED connection, providing the correct orange PED key (PED-authenticated only).

Note: The Crypto User role is initialized by the Crypto Officer. Therefore, only the Crypto Officer, and not the Partition SO, is able to reset the Crypto User credential.

Control the Outcome

The configurable HSM policy 15: SO can reset User PIN allows the Partition SO to control the HSM's response to a set number of consecutive bad Crypto Officer authentication attempts. When the threshold of bad attempts is reached, the CO is locked out of the partiton. If HSM policy 15 is set to 1 (enabled), the partition and its contents can be accessed again after the Partition SO resets the CO’s password. If HSM policy 15 is set to 0 (disabled), then the partition is permanently locked and the contents are lo longer accessible. The partition must be re-initialized and cryptographic material must be restored from backup by the Partition SO.

The configurable partition policy 15: Ignore failed challenge responses can be set by the Partition SO. This policy applies to Activated PED-authenticated partitions only (see "Activation and Auto-Activation on PED-Authenticated Partitions" on page 185). When partition policy 15 is set to 1 (enabled), incorrect partition challenge secret attempts will not apply toward the "failed login attempt" counter.

Resetting Passwords

Resetting is normally done by a higher authority when an authentication secret is lost/forgotten, or compromised, and is discussed separately from merely changing authentication when the user is in legitimate possession of the current authentication.

HSM

There is no provision to reset the HSM SO password (for Password Authenticated HSMs) or the blue PED key (for PED Authenticated or Trusted Path HSMs), except by re-initializing the HSM, which destroys the contents of the HSM and of any HSM partitions. You can change the password (or the secret on the appropriate blue PED key) with the **hsm changepw** command, but that requires that you know the current password (or have the current blue PED key).

The assumption, from a security standpoint, is that if you no longer have the ability to authenticate to the HSM (because you forgot the password or lost the PED key, or because an unauthorized person has changed the password or PED key), then the HSM is effectively compromised and must be re-initialized. Thus, no explicit "reset" command is provided.

The **hsm init** command does not require a login, and the **hsm login** command is not accepted if the HSM is in zeroized state.

For command syntax, see "hsm changepw" on page 1 in the *LunaSH Command Reference Guide*.

Partition

The Partition SO is able to reset the Crypto Officer password or black PED key only if HSM policy 15: Enable SO reset of partition PIN is enabled. By default, this policy is not enabled.

If HSM policy 15: Enable SO reset of partition PIN is enabled and the Partition Crypto Officer is locked out after 10 bad login attempts, then the Partition SO can use the LunaCM **role resetpw** command to reset the Crypto Officer password or black PED key.

For command syntax, see "role resetpw" on page 1 in the *LunaCM Command Reference Guide*.