Luna G5

Administration Guide

THE
DATA
PROTECTION
COMPANY

Document Information

Product Version	5.4.1
Document Part Number	007-011302-009
Release Date	04 July 2014

Revision History

Revision	Date	Reason
A	26 February 2014	Initial release.
B	17 April 2014	Updates to the SFF Backup feature.
C	04 July 2014	Solaris client support.

Trademarks

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 SafeNet, Inc.

Disclaimer

SafeNet makes no representations or warranties with respect to the contents of this document and specifically disclaims any implied warranties of merchantability or fitness for any particular purpose. Furthermore, SafeNet reserves the right to revise this publication and to make changes from time to time in the content hereof without the obligation upon SafeNet to notify any person or organization of any such revisions or changes.

We have attempted to make these documents complete, accurate, and useful, but we cannot guarantee them to be perfect. When we discover errors or omissions, or they are brought to our attention, we endeavor to correct them in succeeding releases of the product.

SafeNet invites constructive comments on the contents of this document. Send your comments, together with your personal and/or company details to the address below.

Contact Method	Contact Information
Mail	SafeNet, Inc. 4690 Millennium Drive Belcamp, Maryland 21017 USA
Email	techpubs@safenet-inc.com

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:

This preface also includes the following information about this document:

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

Audience

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

All products manufactured and distributed by SafeNet, Inc. 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.

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 for this release at the following location:

- http://www.securedbysafenet.com/releasenotes/luna/crn_luna_hsm_5-4.pdf

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.)

Format	Convention
	• 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 ensure that you have read the documentation. If you cannot resolve the issue, please contact your supplier or SafeNet support. SafeNet 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 SafeNet and your organization. Please consult this support plan for further information about your entitlements, including the hours when telephone support is available to you.

Table 1: Technical support contacts

Contact method	Contact	
Address	SafeNet, Inc. 4690 Millennium Drive Belcamp, Maryland 21017 USA	
Phone	United States	(800) 545-6608, (410) 931-7520
	Australia and New Zealand	+1 410-931-7520
	China	(86) 10 8851 9191
	France	0825 341000
	Germany	01803 7246269
	India	+1 410-931-7520
	United Kingdom	0870 7529200, +1 410-931-7520

Contact method	Contact
Web	www.safenet-inc.com
Support and Downloads	www.safenet-inc.com/support Provides access to the SafeNet Knowledge Base and quick downloads for various products.
Customer Technical Support Portal	https://serviceportal.safenet-inc.com Existing customers with a Customer Connection Center account, or a Service Portal account, can log in to manage incidents, get the latest software upgrades, and access the SafeNet Knowledge Base.

CHAPTER 1
Audit Logging

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

- "Overview - Security Audit Logging and the Audit Role" on page 13
- "Configuring and Using Audit Logging" on page 18
- "Audit Logging - Events and Categories" on page 22

Overview - Security Audit Logging and the Audit Role

Beginning with Luna HSM 5.2, Luna HSMs consolidate and enhance auditing of HSM operations.

For Luna PCI and Luna G5 (and the Luna Backup HSM), the audit logging is managed by the HSM Audit role, through a set of lunacm:> 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.

For factory configured Luna HSMs, and after upgrading earlier Luna HSM versions to Luna HSM 5.2, the HSM supports an audit role with authentication via a white Audit PED Key (or an "audit" password for password-authenticated HSMs).

Audit Role on HSM

A Luna 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.

For Luna HSMs with Password Authentication, the auditor logs into the HSM to perform his/her activities using a password, which can be different from the Security Officer (SO) or Partition User passwords, in order to keep the roles separate.

For Luna HSMs with PED Authentication, the auditor logs in to perform his/her activities using a white PED Key. The Audit feature works only with Luna PED version 2.5.0-1 or newer. Older versions of PED firmware are not aware of the Audit role and Audit Key.

Audit initialization - creating the Auditor role (and imprinting the white PED Key for PED authenticated HSMs) does not require the presence or cooperation of the HSM SO.

Audit Role Available Commands

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

```
lunacm:>audit


  The following sub commands are available:

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

  changePw         changePw Change Audit Password
  init             i        Initialize HSM Audit User
  login            logi     Login HSM as Audit
  logout           logo     Logout HSM as Audit
  verify           v        Verify a block of log messages
  config           c        Configure audit parameters
  export           e        Read the wrapped log secret from the HSM
  import           m        Import the wrapped log secret to the HSM
  time             t        Sync HSM time to host, or get HSM time
  status           s        Show status of logging subsystem
  logmsg           logm     Write a message to the HSM's log

  Syntax: audit <sub command>


Command Result : No Error

lunacm:>
```

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

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

Audit Logging

Here is a summary overview of the security audit logging feature:

- Log entries originate from the Luna HSM
- Each entry includes the when, who, what, and result of the logging event
- Multiple categories of audit logging are supported
- Audit management is a separate role - the role creation does not require the presence or cooperation of the Luna HSM SO
- The category of audit logging is configurable by (and only by) the audit role
- Audit log integrity is ensured against -
  - 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
- Certain critical events are logged unconditionally, regardless of the state of the audit role (initialized or not) -
  - Tamper
  - Decommission

- Zeroization
- SO creation
- Audit role creation

Log Origin and Assurance of Integrity

When manufactured, each HSM computes a 256-bit (or 32 bytes) secret random number, called the “log secret”, and saves it on the HSM Flash memory. The log secret is later used to prove that a log record originated from that HSM.

When the HSM needs to log a message, it computes the SHA256-HMAC of all data to be logged, plus the HMAC of the previous log entry, and the log secret. The HMAC is stored in HSM RAM. The record 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 record on the host hard drive. If this is the very first record to be sent to the host ever, then there is no previous HMAC; in this case, the HMAC is set to all zeroes. 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 end-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 in exactly 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 one 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. When checking truncation, the host would send the newest record in its log; and, the HSM would compute the HMAC and compare it to the one in flash. If it does not match, then truncation has occurred.

Log Message Format

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

Offset	Count	Description
0	10	Sequence number
10	1	Comma
11	17	Timestamp
28	1	Comma
29	256	Message text, interpreted from raw data
285	1	Comma
286	64	HMAC of previous record as ASCII-HEX
350	1	Comma
351	88	Data for this record as ASCII-HEX (raw data)
439	1	Newline '\n'

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

The following is a sample log record. It is separated into multiple lines for readability even though it is a one-line 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,
2600001003600B00EA552950140030005D58000003000080010000000000000000000000000000000000000000
```

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 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)”.

The HMAC of previous record is “29C51014B6F131EC67CF48734101BBE301335C25F43EDF8828745C40755ABE25”.

The remainder is the raw data for this record in the form of ASCII-HEX.

Log Rotation Categories, Rotation Intervals, and other Configurable Factors are covered here in the Administration & Maintenance Manual. Command syntax is in the Reference Manual.

Synchronizing Time between HSM and Host

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 audit officer is allowed to synchronize the time.

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 this, the HSM generates a key-cloning vector (KCV, a.k.a the Domain key) for the audit role whenever 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, assuming that HSM is in the same domain, the host passes to the target HSM the wrapped secret, which the HSM subsequently decrypts; any records submitted to the 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 to a separate parameter area for the wrapped log secret.

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.

Capacity

The log capacity of Luna HSMs varies depending upon the physical memory available on the device. The Luna PCI-E HSM and the HSM contained in the Luna SA appliance are the SafeNet K6 HSM card. The HSM inside both the Luna G5 and the Luna [Remote] Backup HSM is the SafeNet G5 HSM module.

The K6 HSM has approximately 16 MB available for Audit logging (or more than 200,000 records, depending on the size/content of each record).

The G5 HSM has approximately 4 MB available for Audit logging (or more than 50,000 records, depending on the size/content of each record).

In both cases, 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. This would be a rare or unlikely event for an HSM connected to a server or workstation, and almost unheard-of in the closed and hardened environment of (for example) a Luna SA appliance.

Time Reported in Log

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

Configuration Persists

Audit Logging configuration is not removed or reset upon HSM re-initialization. It survives tamper and factory reset. 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 most systems, 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 can lose log entries.

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

```
 /usr/safenetlunaclient/bin/pedClient -m stop
```

then

```
/usr/safenetlunaclient/bin/pedClient -m start
```

Configuring and Using Audit Logging

This section describes how to prepare and use audit logging with your Luna HSM.

Required Luna Client version is 5.2 or later; HSM firmware version is 6.10.x or later.

In summary, the steps are:

- Initialize, to create the role on the HSM.
- Configure the various logging parameters.
- Begin collecting and verifying logs of HSM activities.

We also advise that you ensure very reliable timekeeping on the host computer - generally the most reliable option is to use NTP (network time protocol) from a recognized standards organization - and to keep the HSM time synchronized with host time. This ensures that HSM log events and log-file events are in close agreement, which is appreciated by auditing agencies.

If you see the message LUNA_RET_CONTAINER_HANDLE_INVALID this is because you have not yet initialized the "Audit" role on the HSM.

Separation

On a closed, hardened appliance such as Luna SA,with limited user scope, the audit user sees a reduced subset of commands suitable to the audit role, only.

```
Name                    (short)     Description
-------------------------------------------------------------------------------
help                    he          Get Help
exit                    e           Exit Luna Shell
```

```
hsm                      hs          > Hsm
audit                    a           > Audit
my                       m           > My
network                  n           > Network
```

On an uncontrolled host computer, with a contained or attached Luna HSM, all utility commands (lunacm, cmu, ckdemo, etc.) are available to anyone with access to the computer, so a user can see any commands, but can use only those commands that are permitted by the HSM for a specific HSM credential. That is, someone with the audit password (or the white PED Key on a PED authenticated HSM) can use the "audit" commands,but no one else can, including the HSM's Security Officer (the SO). Similarly, the person controlling the audit role on the HSM is unable to use most HSM commands, unless that person also has the SO password (or the blue PED Key for PED-authenticated HSMs). Normally the roles are kept rigorously separate, in order to provide utmost confidence to auditing agencies and to anyone who relies upon their reports and validations.

An instance of lunacm engages a crypto session on the HSM, and then grants user-specific access to HSM functions depending upon the HSM credentials that are supplied. If you are logged into the computer, and using lunacm, and another person needs to access the HSM, you can hand over to them securely in one of two ways:

- explicitly log out of the role that you have been using (SO, audit, Partition User)
  OR
- close lunacm

The first option allows the new person to simply take over your lunacm session, but without allowing them any HSM access that they cannot authenticate for themselves (with the correct password or PED Key). The second option closes the HSM session when the lunacm application closes, which also ends an existing login state. Never walk away from the HSM-containing (or HSM-attached) computer without logging out of any HSM role or closing the utility/application that you have been using.

Detailed steps

1. Before configuring audit logging for a Luna HSM, first ensure that the PedClient (also called the callback server) has been started. If the callback server is not started, "audit" commands will return CKR_CALLBACK_ERROR.
2. Run the command:
   lunacm:>audit init
   For password-authenticated HSM, you are prompted for a domain string or password; for PED-authenticated HSM, you are referred to Luna PED, which prompts for a white PED Key.
3. Now that the Audit role exists on the HSM, the auditing function must be configured. However, before you can configure you must authenticate. Run the command:
   lunacm:>audit login
   For password-authenticated HSM, you are prompted for a domain string or password; for PED-authenticated HSM, you are referred to Luna PED, which prompts for a white PED Key.
4. When your credentials have been accepted run :
   lunacm:>audit config
   The first time you configure, we suggest using only the "?" option, in order to see all the available options in the configuration process. For example,
   lunacm:>audit config eventmask l
   will log everything the HSM does. This might be useful in some circumstances, but will quickly fill up log files. The addition of,
   lunacm:>audit config i h@15
   would rotate the logs on an hourly interval, at the 15-minute mark of each 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.

5. Specify the audit log path on the host computer :
   ```
   lunacm:>audit config p /usr/safenet/lunaclient/log
   ```
   In this case, "log" is a directory/folder name. and must NOT be a filename.
   The system specifies each filename - attempting to set a path that includes a filename would result in CRK_LOG_BAD_FILE_NAME.

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 host grows according to the logging settings and the rotation schedule that you configured (above). At any time, you can copy files to a remote computer and then clear the originals from the host, if you wish to free the space.

Export the Audit Logging secret from the HSM and import to the verifying HSM

1. On the host computer where HSM audit log files are being created, export the Audit Logging secret:
   ```
   lunacm:> audit export
   ```

2. Exit lunacm and browse to see the filename of the wrapped log secret.
   ```
   /user/safenet/lunaclient/bin :>cd ../../lunalog
   /user/safenet/lunalog :>ls
   123456 7001347 k6secret.bin  LogSecret_130115210057_123456.lws
   ```

3. On the computer where the HSM is attached, that you will use to verify the downloaded Audit Log file, run:
   ```
   /user/safenet/lunaclient/bin :>scp audit@myhost1:/usr/safenet/lunalog/LogSecret_130115210057_123456.lws .
   ```
   (substitute the actual file name of the exported secret in the above example command) and provide the audit user's credentials when prompted. This copies the identified file from the remote host computer'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.

4. Launch lunacm,
   ```
   /user/safenet/lunaclient/bin :>./lunacm
   ```
   For this example, we will assume that you have initialized the HSM Audit User role, using the same domain/secret as is associated with the source Luna HSM.

5. Import the Audit Logging secret into the locally attached HSM:
   ```
   lunacm:>audit import file 151170.lws
   ```

6. Verify the file
   ```
   lunacm:>audit verify file mylunsa1_audit_2013-02-28.tgz
   ```

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

Additional Considerations

1. 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. This is equivalent to the same situation for the HSM's Security Officer (SO). The following examples illustrate some points of behavior.

   Example #1: SO or audit role is not initialized, no login is needed

   ```
   lunacm:>hsm init -l myHSM -d default -p userpin -f
   'hsm init' successful.

   lunacm:>audit init -d default -p userpin -f
   Command Result : 0 (Success)
   ```

   Example #2: SO or audit role is initialized, but not logged in

   ```
   lunacm:>hsm init -l myHSM -d default -p userpin -f
   Error:  'hsm init' failed. (1010000 : LUNA_RET_USER_NOT_LOGGED_IN)
   lunacm:>audit init -d default -p userpin -f
   The audit sub-command failed. (LUNA_RET_USER_NOT_LOGGED_IN)
   ```

   Example #3: SO or audit role is logged in, init with the correct password and new domain

   ```
   lunacm:>hsm init -l myHSM -d safenet -p userpin -f
   'hsm init' successful.

   lunacm:>audit init -d mysafenet -p userpin -f
   Command Result : 0 (Success)
   ```

   Example #4: SO or audit role is logged in, init with the wrong password

   ```
   lunacm:>hsm init -l myHSM -d safenet -p wrongpin -f
   Error:  'hsm init' failed. (A00000 : LUNA_RET_UM_PIN_INCORRECT)
   lunacm:>audit init -d default -p wrongpin -f
   The audit sub-command failed. (LUNA_RET_UM_PIN_INCORRECT)
   ```

2. Multiple bad logins produce different results for the SO and 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_SO_LOGIN_FAILURE_THRESHOLD error is returned, but the HSM is unaffected. If 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 will be successful.

3. In the event that the current audit log file is locked or corrupted on a Luna PCI-E or a Luna G5 host, for example due to a system crash, the audit logger might enter a state where it would repeatedly try and fail to open the current audit log file. Any audit re-configuration attempt might also fail. Follow the procedure below to get out of the situation:
   - Stop the pedClient (also known as callback server), which serves as the audit logger
   - Move the current audit log file to the ready_for_archive folder or directory
   - Start the pedClient

   You might not be able to verify the corrupted log file. All other log files should be verifiable.”

Audit Logging - Events and Categories

This section summarizes the audit logging events by category.

HSM Access Events

HSM Events	Descriptions
LUNA_LOGIN *	C_Login
LUNA_LOGOUT *	C_Logout
LUNA_MODIFY_OBJECT	C_SetAttributeValue
LUNA_OPEN_SESSION *	C_OpenSession
LUNA_CLOSE_ALL_SESSIONS	C_CloseAllSessions
LUNA_CLOSE_SESSION *	C_CloseSession
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 Events

HSM Events	Descriptions
LUNA_LOG_EXTERNAL	CA_LogExternal

HSM Management Events

HSM Events	Descriptions
LUNA_ZEROIZE **	CA_FactoryReset

HSM Events	Descriptions
LUNA_INIT_TOKEN **	C_InitToken
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	Luna 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	Luna internal, for manufacturing
LUNA_MTK_UNLOCK_CHALLENGE	Luna internal, for manufacturing
LUNA_MTK_UNLOCK_RESPONSE	Luna internal, for manufacturing
LUNA_MTK_RESTORE	CA_MTKRestore
LUNA_MTK_RESPLIT	CA_MTKResplit
LUNA_MTK_ZEROIZE	CA_MTKZeroize
LUNA_FW_UPGRADE_INIT	CA_FirmwareUpdate
LUNA_FW_UPGRADE_UPDATE	CA_FirmwareUpdate

HSM Events	Descriptions
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 Events	Descriptions
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
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

HSM Events	Descriptions
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 Events	Descriptions
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
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

HSM Events	Descriptions
LUNA_WRAP_CSP	CA_CloneMofN_Common
LUNA_M_OF_N_DUPLICATE	CA_DuplicateMofN
LUNA_ENCRYPT_SINGLEPART	C_Encrypt
LUNA_DECRYPT_SINGLEPART	C_Decrypt
LUNA_PE1746_COMMAND	Used when PE1746 is enabled

Audit Log Management

HSM Events	Descriptions
LUNA_LOG_SET_TIME	CA_TimeSync
LUNA_LOG_GET_TIME	CA_GetTime
LUNA_LOG_SET_CONFIG *	CA_LogSetConfig
LUNA_LOG_GET_CONFIG *	CA_LogGetConfig
LUNA_LOG_VERIFY	CA_LogVerify
LUNA_CREATE_AUDIT_CONTAINER **	CA_ InitAudit
LUNA_LOG_IMPORT_SECRET	CA_LogImportSecret
LUNA_LOG_EXPORT_SECRET	CA_LogExportSecret

*: The event must be allowed to proceed even if the result should be logged but cannot (for example, due to a log full condition).

**: The event is logged unconditionally.

CHAPTER 2

Backup and Restore

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

Backup and Restore in General

Backup of Luna G5 HSMs uses the cloning feature and the `hsm clone` or `partition clone` commands. Cloning takes place from hardware to hardware (from one Luna G5 HSM connected to your computer to another Luna G5 HSM connected to the same computer) in secure fashion.

When backing up the contents of an HSM or a partition on an HSM, the source and target HSMs must share the same cloning domain (red PED Key for PED Authenticated HSMs), if cloning is to take place. The domain is set at initialization time, and cannot be altered without initializing the HSM again. During the transfer, all data is encrypted with the domain secret.

HSM or Partition

Roughly equivalent backup and restore options exist for the main SO space of the HSM, and for the User Partition. These are handled separately.

For the HSM level, if you clone to a second HSM you wipe that target (initialize it) and fill it with the SO objects from the source HSM. In most cases there would not be any SO-only objects - you would be cloning the structure (authentication, etc.) Similarly, if you restore to an HSM, you initialize it in the process. This means that you cannot incrementally or selectively restore HSM SO-owned objects via cloning, so you cannot keep any changes that you made in the original HSM after the backup (clone) was made.

Normally, this is not an issue, because there is usually little need to backup SO-owned objects at the HSM level. The SO level is usually only administrative on the HSM.

The more usual requirement is to backup the working contents of a User Partition, which is the level where the real work of your Client applications takes place, and the working keys and certificates and other objects are stored.

For partitions, the cloning can include all partition objects or a subset that you indicate by a (comma-separated) list of object handles.

Your backup and restore operations are "lunacm partition clone", in either direction (**to** the target HSM's partition for the backup operation, or **from** it for the restore operation).

Other Info

The authentication for the Backup Tokens can match that of the HSM or Partition, or it can be different. This is a decision that should be referred to your organization's security policies. However, the HSM and the backup token must share the same domain.

Backup Your HSM Contents

You can backup the non-User (non-partition) objects on your HSM - which would be either public objects or objects owned by the SO - with the "clone" feature.

Cloning Backup to another HSM

HSM cloning securely clones HSM objects (not including objects that are contained within HSM Partitions), from the HSM to another HSM in your computer. To backup the HSM, have ready a blank HSM, or one that is acceptable to re-initialize (initializing, or re-initializing an HSM destroys any material that was on the HSM).

To backup your HSM

1. Have two HSMs connected to your computer.
2. Start the lunacm utility.
3. Login to the primary/source Luna HSM as SO.
4. At the lunacm prompt, type :
   ```
   hsm clone -objects <handles> -slot <slot number> [-password <password>]
   ```
   (the '-password <password>' is needed only if your HSMs are Password Authenticated
   the **source HSM is the current slot** while the target HSM is the slot that you indicate in the command).
5. Secure the receiving/target HSM. Best practice for important keys and objects is to have a backup HSM in onsite secure lockup, for quick resumption of service in case of damage or loss of the primary HSM, and another backup HSM in secure off-site storage for disaster recovery.

To later restore the SO's token objects, perform a cloning operation from the backup token to the HSM that needs the objects.

See "Backup (Clone) Your HSM Partition" on page 28 for separate handling of partition objects.

Additional Notes

Backing-up/cloning the SO-space to a target requires that the target HSM be initialized as part of the process.

If there are no SO objects to clone (a common situation), then the process halts.

Backup (Clone) Your HSM Partition

As described elsewhere, you can have a Luna HSM:

- with Cloning capability (direct, secure transfer from one HSM to another)

"**Partition clone**" securely clones partition objects (not including SO objects that are contained on the HSM, but not within an HSM Partition) from the HSM Partition to an encrypted file on your computer. The two HSMs must share the same domain (that is, they must have been initialized with the same red PED Key (for PED authenticated version) or the same text Domain value (Password authenticated versions)).

To backup your HSM partition

1. Start the lunacm utility.
2. Login to the partition as User.

3. At the lunacm prompt, type
   ```
   partition clone -objects <handles> -slot <target slot> [-password <secret>] [-
   force]
   ```
   (the '-password <password>' is needed only if your HSMs are Password Authenticated
   the **source HSM is the current slot** while the target HSM is the slot that you indicate in the command).

4. Secure the receiving/target HSM. Best practice for important keys and objects is to have a backup HSM in onsite secure lockup, for quick resumption of service in case of damage or loss of the primary HSM, and another backup HSM in secure off-site storage for disaster recovery.

CHAPTER 3

High Availability (HA) Mode

This chapter describes how to configure and use your HSMs in high-availability (HA) mode. It contains the following sections:

About Luna G5 HA Groups

For Luna G5 , an HA Group is an identified group of Luna G5 HSMs attached to one workstation or server. The purpose is to provide load balancing among HSMs connected to one computer. The members of the group are identified in the HA section of the Luna configuration file.

A single Luna G5 HSM can perform about 200 RSA 1024-bit signings per second (assuming ideal conditions and 10 threads). Two Luna G5 HSMs, configured as an HA group, are good for about 400 signings per second in the same conditions. For asymmetric algorithms, there is little overhead. For symmetric operations, some overhead is noticeable, but performance scales with additional units, regardless.

Luna G5 HA creates a virtual slot against which your application can run, instead of the application making direct use of the individual HSMs. This can be further emphasized by setting the "HAOnly" option, which hides the physical HSMs from your application, presenting only the virtual slot. Your application does not need to know about physical HSM slots - identity, the number that are present, etc. - it directs all calls to the virtual slot, and receives all responses from the virtual slot.

Limitations

Luna G5 HA is not High Availability as it is usually understood. Failover does occur - a failed member that is restarted does automatically rejoin the group - however, because both/all HSMs in the group are connected to the same computer, that single host is the point of vulnerability.

Individual HA Group members are unaware of each other. HA is controlled at the Luna Client software level.

A Luna G5 HA group cannot include HSMs from more than one host computer.

Usage Notes

HA group membership is tracked through the configuration file, but the order of appearance carries no meaning, other than that the primary is the first HSM assigned a call from the newly started client application. Once the application has been in operation against the virtual slot, with physical-slot activity being assigned on a round-robin, least-busy basis, any of the physical slots can be the first to experience a change (addition, deletion of objects) which is then cloned to the other physical slots in the process of synchronization. In normal operation, there is no hierarchy of physical HSMs.

Applications can see all slots - the virtual slot and the physical slots (the HSMs) - and perform actions upon them, unless the "haonly" option is set (which hides all but the virtual slot). However, only object creations and deletions that are ordered at the virtual slot are automatically synchronized to all physical members of the group. If an application addresses a physical slot directly (bypassing the virtual slot) and creates an object, that object becomes an "orphan", residing on just that one slot. The HA group continues to function, but it is unsynchronized. If you (your application) send a call to the virtual slot that needs the orphan object(s), the result could be an error message, since the virtual slot has not been made aware of the objects and has not replicated them across the group.

If you wish to replicate such an "orphan" object, perform synchronization manually.

You can perform such "orphan" actions without problem as long as you ensure that calls to use the object are directed to only the relevant physical HSM. All other calls to the HA group can continue via the virtual slot.

The recommended method is for your application(s) to always deal with the virtual slot, ensuring that synchronization occurs automatically.

HA Operational Notes

When your application is using a Luna G5 HA group, your application appears to be using just one HSM – the virtual or group HSM that hides the HA group members. Your client should not attempt to directly address any partition on any Luna G5 within the HA group. This defeats the purpose of HA, and can cause disruption if you/your application attempts to change anything on just one member of a synchronized group. Similarly, no other application or user should be permitted to address any of the HA group members individually. As long as your application addresses its requests to the virtual group Partition, the HA functionality takes care of all activity in transparent fashion.

The intent of Luna G5 HA at this time is to provide :

- load balancing
- operational redundancy such that if aunit fails (or must be taken off-line for other reasons) the remaining HSMs can continue to provide service to the Client application until the failed/removed HSM (or a replacement unit) can be brought into the HA group.

Load Balancing

Load balancing is supported for single-part operations like sign, verify, encrypt, decrypt. For multi-part operations, the operation is performed in the primary Luna G5's partition and the results are then cloned to the other member(s) of the HA group.

Reconnecting an Off-line Unit

- In HA mode, if a Luna G5 goes off-line/drops-out (due to failure, maintenance, or other reason), the application load is spread over the remaining Luna G5 Partitions in the HA Group.
- When the unit is restarted, the application does **not** need to be stopped and restarted, before the re-introduced unit can be used by the application.
- For the unit that was withdrawn (or for a replacement unit), you must re-Activate the Partition before it can be re-included into the HA Group.

Replace a Failed Group Member with a New Luna G5

1. Configure the new Luna G5, making it part of the same cloning domain as others in the HA group (at initialization, get its cloning domain from the same red domain PED Key).
   If you require that the replacement appliance must have the same name as the replaced appliance, then you will need to stop your application before introducing the new appliance.
2. Create a partition with the same characteristics as others in the HA group .

3. Determine the serial number of the failed member partition.
4. Remove the failed member from the HA group using the "lunacm" command:
   ```
    lunacm:> haGroup -removeMember -group <groupNumber> -serialNum <serialnumber> -
   password <password>
   ```
5. Add the new partition of the new Luna G5 to the HA group using the "lunacm" command:
   ```
    lunacm:> haGroup -addMember -group <group number> -serialNum <serialnumber> -
   password <password>
   ```
6. Perform a manual re-synchronization between the members using the “lunacm” command:
   ```
    lunacm:> haGroup -synchronize -group <GROUP NAME>
   ```

Upgrading and Redundancy and Rotation

The Luna G5 HA function assumes that all Luna G5 appliances in an HA group are at the same appliance software and firmware level. Therefore, when you intend to upgrade/update any of the Luna G5 units in an HA group, or when you intend to upgrade/update the Luna G5 Client software, schedule some downtime for your application.

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 uptime-critical applications, you would likely already have such a backup set of Client-plus-HA-group that you would rotate in and out of service during regular maintenance windows.

Using Algorithms or Features in a Mixed HA Group

While it is possible to have HSMs with different firmware versions within an HA group, this is not generally recommended. Be aware that the capability of the group (in terms of features and available algorithms) is that of the member with the oldest firmware.

For example, if you had an HA group that included an HSM with two different firmware versions, then certain capabilities that are part of the newer firmware would be 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 was load-balanced to an HSM that did not support the newer features it would fail. Similarly, if the newer-firmware HSM dropped out of the group, operations would fail. Your Clients must not invoke those algorithms because not every member of the group supports them. The solution is to upgrade the older units to the most recent firmware and software versions (where possible) or else to limit clients to only the lowest supported feature set.

Frequently Asked Questions

If a partition becomes full, what happens?

You can't create any more objects on it. Some scenarios are just what they seem and have no bearing on HA, in particular...this is one of them.

Are session objects replicated or only token objects?

Session objects, as well as token objects, are synchronized and replicated.

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. 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 resent from the calling application. That is, if you don’t receive a ‘success’ response, then you must try again. This is obviously 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.

How many times, or for how long will a device be polled to be automatically reintroduced?

This is set when you enable the feature. You can try once per minute, up to 500 minutes.

How does the automatic reintroduction work? Why does it need a partition policy?

Logic is built into HA client code.

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.

Under what circumstances will a device be moved out of an HA group - only in the event it cannot be contacted?

You must manually remove a member. If the device cannot be contacted, the HA client merely stops trying it (see "retries" in the previous question), but the device remains a group member until manually removed.

Can you add and remove devices to a HA group without restarting the application? If so what caveats apply?

No, you cannot. Think of starting the application as starting a race. You cannot add in a new runner once the race is already under way. But, if you restart the race, you can.

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.

If an HA group member fails and an application restarts, it will not be possible to recover that device until you restart the application again. Why?

This is as designed. You originally had your application running with X number of members. One failed, but was not removed from the group, so retries were occurring, but the application was operating with X-1 members available. Then you restarted. When the application came up after that restart, it saw only X-1 members. Having just started, it now has no notion that the Xth member exists. The "race" has restarted with X-1 runners. You cannot add to that number within an application. To go from the number that the application now recognizes, X-1, to the new, larger number of participants X-1 +1 (or X), you must restart the application (the race) while all X members (runners) are available.

Can a PED operation on one member of an HA group lock it out from operation (PED operations block cryptographic operations)? If so, will it automatically come back into use after the operation has concluded?

Yes. Fail-over and recovery HA logic are invoked.

What if HA does not recognize partition full?

Normally, this could happen only if you are performing operations directly on physical slots, rather than via the virtual slot. 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.

CHAPTER 4

Hardware Maintenance and Configurations

This chapter describes tasks related to maintaining and configuring the Luna G5 hardware. It contains the following sections:

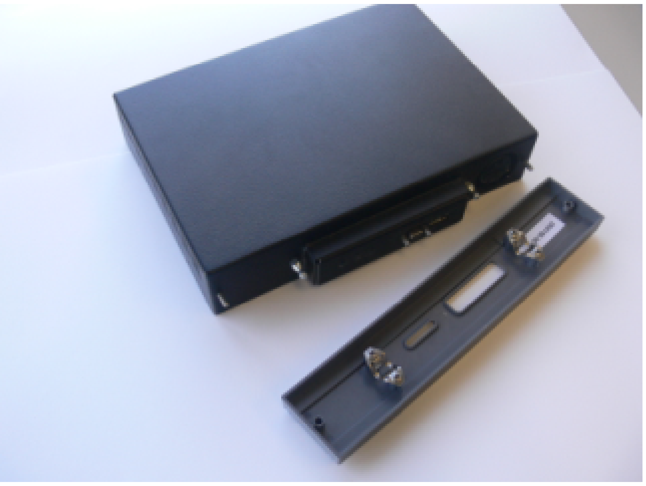
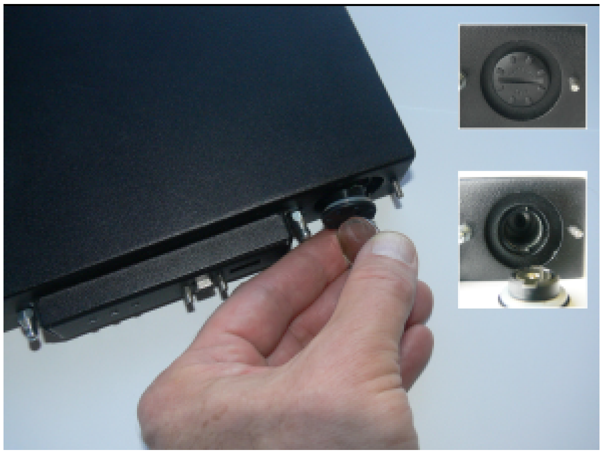
- "Luna G5 HSM Battery Installation" on page 35
- "Luna G5 HSM Battery Questions" on page 38
- "Connecting Multiple Luna G5 Units to One Computer" on page 40

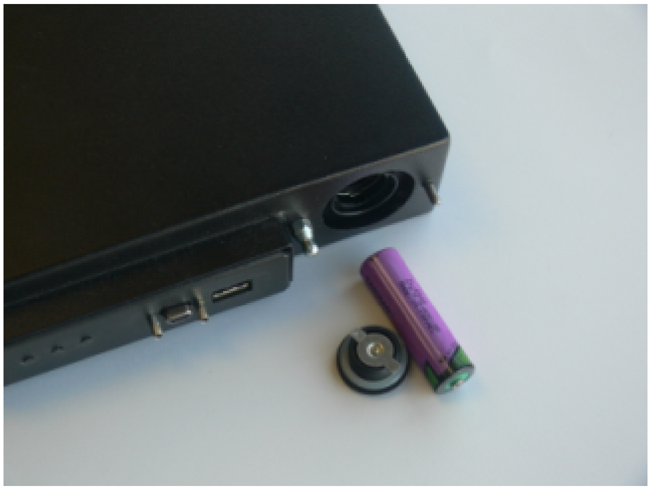
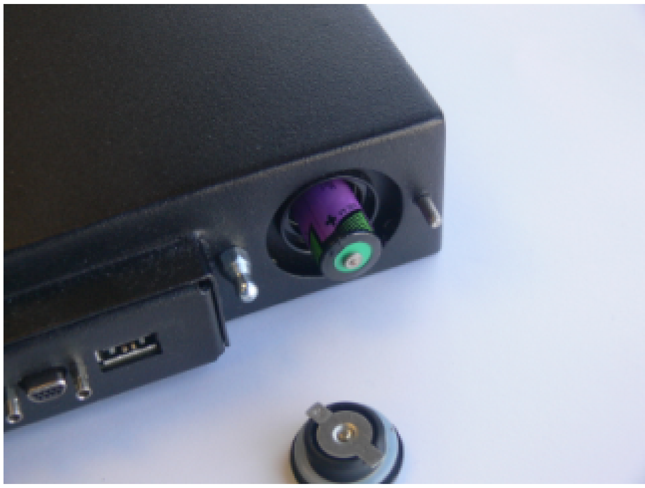
Luna G5 HSM Battery Installation

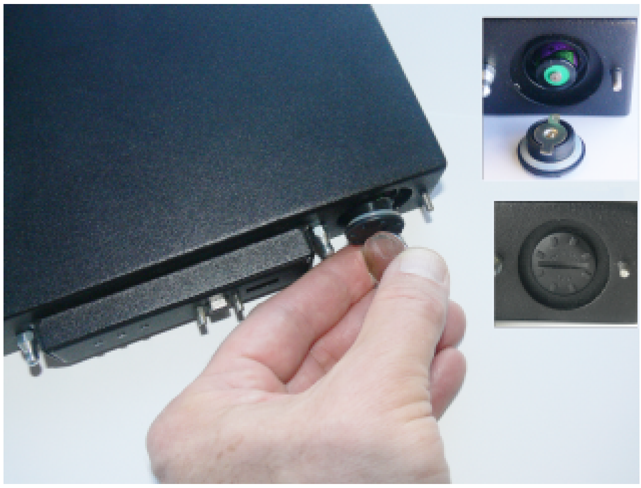
The Luna G5 HSM comes from the factory with its battery packed separately. The battery must be installed before you can configure or use the HSM.

The battery that powers the NVRAM and RTC inside the HSM is shipped in the packaging, but outside the Luna G5 HSM.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 external HSM to place it into service, then you must install the battery before using the device. Here are the instructions (as also seen in the Luna G5 Quick Start Guide).

To install the battery

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.

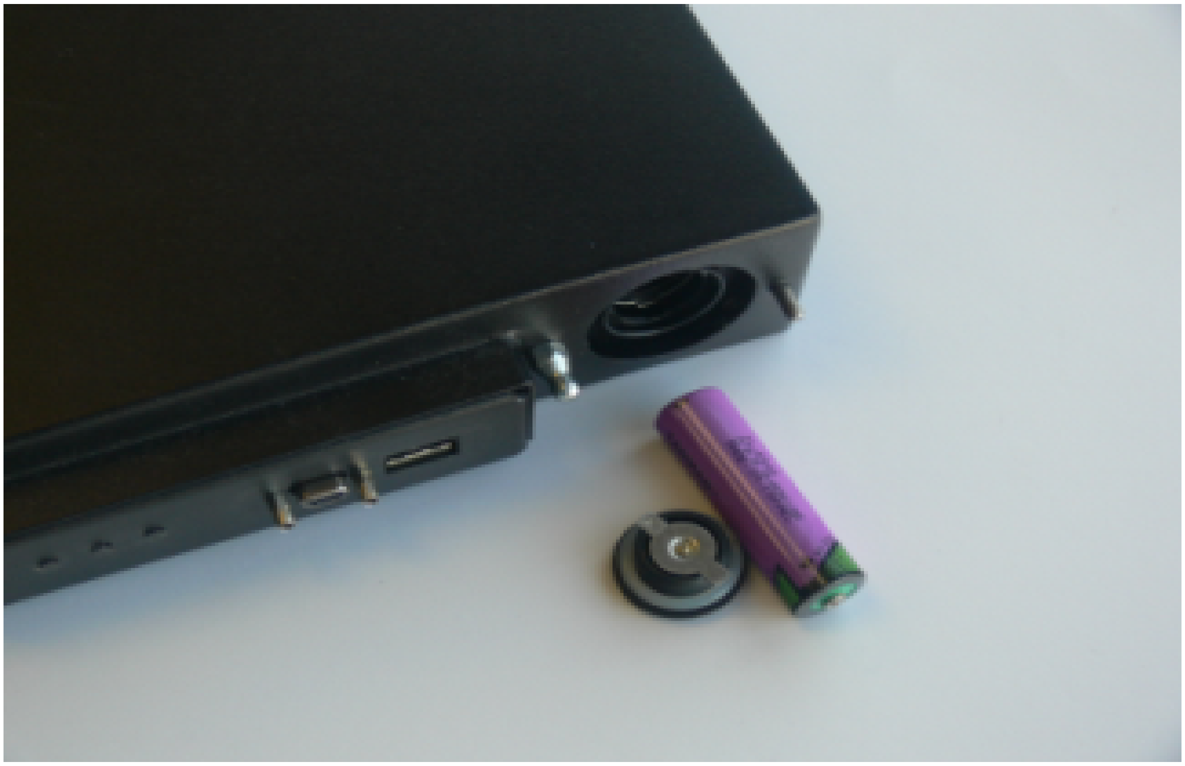
3	 Remove the battery from its packaging and align it at the opening of the Luna G5 battery compartment. The battery has a “+” sign near the end with the raised nub/bump. The flat end of the battery is the negative pole (-).
4	 Insert the battery, negative end first. The positive end (+) should protrude. 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 Luna G5. 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. Connect and configure your Luna G5 HSM .

.

Luna G5 HSM Battery Questions

The Luna G5 HSM can be stored, with valuable contents, when not in use.



The battery that powers the NVRAM and RTC must be installed for use, but some questions commonly arise if the device is to be stored for long periods. As an administrator of HSMs, I need clear instructions on what to do/how to manage the battery in the Luna G5 and Luna Backup HSM so that I don't get into a situation where I can't retrieve my backups or use my HSM.

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

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

If the battery is out, what happens?

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

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

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

There is no corruption of stored objects.

Where can I get a spare/replacement battery?

From any supplier that can match the specifications.

Technical Specs:

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

Fast voltage recovery after long term storage and/or usage

Low self discharge rate

10 years shelf life

Operating temperature range -55 ºC to +85 ºC

U.L. Component Recognition, MH 12193

Storage Conditions:

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

Temperature should not exceed +30 ºC

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

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

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

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

If your HSM is a Password-authenticated version,
OR
if your HSM is a PED-authenticated version, but you have not moved an MTK split out of the HSM (onto a purple SRK), then simply insert the battery, connect the HSM, power it up, and resume using it.

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

If your HSM is a PED-authenticated version, and you have previously enabled SRK (moved one split of the MTK out of the HSM, onto a purple PED Key - the SRK), then the first time you attempt to use the HSM (after battery replacement and power-up), the HSM is unable to find the "missing" portion, in order to reconstitute the MTK. You are prompted to present the purple PED Key. As soon as the correct SRK is received, the MTK is reconstituted, and all your stored material is available for use.

Connecting Multiple Luna G5 Units to One Computer

Luna G5 is a USB device; therefore multiple units can be connected and working simultaneously at one computer.

One limitation would be the number of USB ports on your computer.

If you need to exceed the number of USB ports directly available on your computer, be sure to use a high-quality USB hub. Generally, we suggest not using the external hubs that are integrated into some monitors and keyboards - or you can try such a connection if it is convenient, and if you encounter problems with your Luna G5, then reconnect it more directly to your computer.

Considerations

When connecting multiple Luna G5 units (to a computer that already has the software and driver installed), wait 30 seconds between connections, to allow each unit to perform its onboard self-test and to complete its handshake with the computer.

If you have multiple units connected and must power-cycle your computer, the USB-connected devices will receive a reset. Therefore, we suggest that you disconnect all but one Luna G5 and then, after the computer has completed its startup and recognized the single connected Luna G5, connect the other units at 30-second intervals until all have been reconnected and are once again working.

If a power outage occurs, involving a computer and multiple connected Luna G5s, verify upon return from the outage that all Luna G5s are properly recognized (the slots show the correct identification and you can see the objects in the HSM partitions). If there is any problem, try disconnecting and then reconnecting the Luna G5 units on 30-second intervals.

CHAPTER 5

Re-initialization and Zeroization

This chapter describes how to re-initialize a previously configured HSM, and the impact of the re-initialization. It contains the following sections:

- "HSM Initialization and Zeroization" on page 42
- "Re-initialize an HSM" on page 42

HSM Initialization and Zeroization

Ideally, the 'hsm init' command is used once, when you first configure your Luna HSM for use with your application, then you place the unit in service and never initialize it again. However, unanticipated situations or requirements can arise that might cause you to initialize the HSM. A simple example is that you might perform trial setups in a laboratory environment before placing your Luna system into a "live" or "production" environment.

"Soft" Init

If initialize (`hsm init`) is called when the hsm is not zeroized, the SO is required to login (must present current SO PED Key or hsm password to authenticate for the init command). The firmware erases all partitions and all SO objects. However, the cloning domain and hsm policy settings (any of which are applicable) all remain unchanged.

"Hard" Init

If the hsm is zeroized when the init call is made, the firmware performs a full initialization, including: set SO pw, set domain, set M of N.

Additional Notes

The lunacm command 'hsm factoryReset' puts the HSM in a zeroized state. To completely start over for configuration of the HSM, use 'hsm factoryReset', then 'hsm init'.

It is not necessary to perform 'hsm login' before 'hsm factoryReset'. This is not considered a security issue because, if the application and your secured data (keys, certificates, etc.) are critical, then you would necessarily ensure the physical security of any computer where the HSM is used, and have your data safely backed up. In other words, anyone who can gain physical access to the HSM, and issue the 'hsm factoryreset' command without your authorization, is only destroying the HSM contents - not viewing them or altering them - which they could also achieve by inflicting physical damage (given that you have permitted them to reach the HSM anyway, in which case you have real security concerns to address).

Re-initialize an HSM

To initialize or re-initialize an HSM, use the command:

```
lunacm hsm init -label <new-HSM-label>
```

Note: Initializing/re-initializing an HSM destroys all HSM Partitions, and all contents are lost. This is not an action you would perform on a production Luna HSM. However, if you have made major changes in your system/deployment, or if you are moving a Luna HSM from a lab situation into production, you might wish to clear everything and restart with a "clean slate". In such cases, re-initialization might be appropriate. It would also be appropriate if you were so instructed by Customer Support.

Note: Invoking the lunacm hsm init command results in the HSM Admin/Security Officer being logged out, MofN being deactivated (applies to specific configurations only), and all partitions being deactivated. These preparatory actions take place before the warning prompt appears, with its request for you to type "Proceed" or "Quit". That is, if you invoke luna cm hsm init and then type "quit" at the prompt, initialization does not take place (meaning that you do not lose existing token/HSM contents), but any current login or activation state is closed, whether you abort the command or not.

CHAPTER 6

Key Migration

This chapter describes how to migrate key material from one HSM to another. It contains the following sections:

- "Key Migration Procedures" on page 44
- "Frequently Asked Questions" on page 44

Key Migration Procedures

If you have other Luna HSMs on which you have important keys or data, you can securely migrate that material to another Luna HSM. Contact SafeNet Technical Support and ask for the following document:

- 007-011528-001 *Luna HSM Key Migration Guide*

Frequently Asked Questions

We want to generate keys on one HSM and copy them to other HSMs. Can they have the same object handles?

No. You can clone keys between HSMs that share a domain, but each HSM assigns its own object handles to incoming - or generated - objects.

Good PKCS#11 applications **never** make assumptions about the object handle number.

Typically, an application will find an object prior to use; for example, find by CKA_LABEL is the most common.

The label either is known to the user or is published somewhere application-specific; for example, Microsoft uses the certstore to store the label (a.k.a. container name).

Possible workarounds:

If your application already uses handles to access/identify keys, consider identifying keys by fingerprint (and possibly label) and devising your own mapping to the new handles for objects that you import (clone) into the HSM.

HOWEVER, that approach might not be feasible if you are not in a position to make API changes - such as, if you are using a third-party application, or if you are locked in by internal compliance/audit or by external compliance/audit. Then, perhaps you could consider using multiple HSMs in an HA group.

If you are accessing via an HA group, then the HA group has a single virtual handle for each object that your application would see, regardless of the "real" object handle on each HSM.

CHAPTER 7

Partition Management

This chapter describes the tasks associated with managing the user partition on the HSM. It contains the following sections:

- "About Activation and Auto-Activation" on page 45
- "Creating and Changing Partitions and Users" on page 47

About Activation and Auto-Activation

Client access to Partitions, on a Luna HSM with PED (Trusted Path) Authentication, needs to be as efficient and convenient as Client access to a Password Authenticated Luna HSM . Activation and autoActivation are ways to manage the additional layer of authentication - the Luna PED and PED Keys, so that Clients can reliably connect using just their passwords.

Activation caches the partition black PED Key authentication data (on the HSM) for the duration of the current session.

Autoactivation caches the partition data and preserves it through a brief power outage. Autoactivation applies only to standalone Luna HSM appliance products where the battery provides the power to the cache memory during a power outage (up to 2 hours).

The General Case, without Activation

When a command is issued to the Luna HSM that requires HSM or Partition authentication, the Luna HSM with PED (Trusted Path) Authentication looks to the PED. Luna PED responds by prompting you for actions involving the appropriate PED Keys and the Luna PED touchpad. If Luna PED gets the appropriate response, it confirms the authentication back to the Luna HSM, via the PED interface (the Trusted Path). The required PED Keys would be:

- the blue key needed when the Security Officer or HSM Admin logs in, or issues an `hsm` command.
- the black key (possibly several black PED Keys, but only if you invoked M of N when you initialized the HSM), needed when the Partition Owner (or Crypto Officer, if you are working under that model) issues Partition administration commands, or creates, deletes (or otherwise manipulates) non-public objects.

Those PED Keys (as appropriate), are demanded by Luna PED when you perform administrative operations via the `lunacm` interface. The authentication can consist of:

- presenting the required PED Key(s) and pressing [ENTER] on the touchpad, or
- presenting the required PED Key(s), pressing [ENTER], entering a PED PIN (if one had been assigned at initialization) and pressing [ENTER] again

Performing the above actions gets you to a login state in which Luna HSM will carry out HSM or Partition commands (according to the level of authentication that you invoked). User challenge text strings are still required, in addition to the black PED Key, when the Crypto Officer [the partition administrator] or Crypto User [the operational entity that runs application programs to use the HSM's objects and services] wishes to perform actions in the partition.

Activation

Activation is just a login with explicit caching of the Partition black PED Key login data, on the Luna HSM. This is convenient so that you can remove the black PED key (perhaps to allow other uses of the Luna PED, such as administrative logins by the HSM Admin or SO), while ensuring that access by Clients is not stopped, and that no-one is required to be present to press [ENTER] on the touchpad for the benefit of Clients.

Note: For the Crypto Officer and Crypto User, a challenge secret MUST be created in order for activation to work. If you are familiar with some other Luna HSM products (like Luna SA) the Luna G5 HSM differs slightly, such that the challenge is not automatically created when the user is created. You must run the **createChallenge** command (and record the resulting challenge secret/password for each entity, to be able to provide it when that entity needs to use the activated HSM partition).

We emphasize that it is the authentication data from the PED Key(s) that is cached, because the partition challenge secret - or optionally the Crypto User challenge secret - is not cached. That secret must still be supplied by your application in order to make use of objects on the partition. Thus, you have the challenge secret (the one that you recorded from the screen of the Luna PED during Partition creation or User creation) embedded in your Client application. The Client can make use of that challenge only if the Partition is in login state. That can be an explicit login by somebody who is present and ready to act with the PED Key(s) and any PED PIN (optional), or it can be the presence of cached login data (Activation) which the HSM looks for every time a client access request comes in.

To use Activation, you must first allow it by setting Partition Policy 22 (Enable Activation) to *on*, for each Partition that you create. If the Policy (22, Enable Activation) is on, then the Partition Owner (or Crypto Officer) can issue the `lunacm partition activate` command. Luna PED prompts for the black PED Key and any other authentication input that might be appropriate (PED PIN, M of N). Once you provide it, the Luna HSM caches that authentication and the Partition remains in a login state (Activated) until:

- you explicitly deactivate (with lunacm command `partition deactivate`), or
- power is lost to the Luna HSM.

After activating, you can remove the black PED Key and keep it in your pocket or in safe storage. Activation remains on, and any registered Client with the Partition Password (challenge secret) is able to perform operations on the Partition.

AutoActivation

When AutoActivation is added to Activation, the User authentication data is cached in such a way that it can survive a power outage of approximately 2 hours in duration.

Ensure that the Partition Policy number 23 "Allow auto-activation" is switched on (set to 1).

When that policy is on (and policy 22 as well) then whenever you issue the lunacm command "partition activate", the User authentication data is cached for auto-activation.

Deactivating a Partition

You can turn off Activation for an HSM Partition by issuing the deactivate command.
Type:

```
lunacm:> partition deactivate
```

The User's (black) PED Key data is de-cached. On HSM types with a single partition, the command has no options or arguments. The next time access to the Partition is attempted (to run a lunacm partition command or to perform cryptographic operations on behalf of the client application), you are directed to the Luna PED, to provide the black PED Key (and PED PIN if applicable).

Creating and Changing Partitions and Users

A partition and its users are created when you configure the HSM. To destroy an existing HSM Partition, issue the lunacm `partition create` command. The difference between creating a partition the first time and creating a partition where one already existed is the warning from lunacm, so that you do not inadvertently destroy a valuable partition.

```
lunacm:> partition create
The existing Partition will be destroyed.
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
lunacm:>
```

For the PED Authenticated HSM version, as shown above, the Luna PED gives you the usual opportunity to imprint a new Partition authentication secret on a new black PED Key, or to overwrite an existing secret on a PED Key, or to accept an existing secret (only if it is a Partition secret - existing HSM SO or Domain secrets are not accepted for this purpose).

For the Password Authenticated version, obviously there would be no mention of a PED, and you would be prompted to supply a new Partition password.

Users

On a PED Authenticated HSM, the **User** or **Owner** [- the term User is a standard PKCS#11 nomenclature, from the days before HSMs, while the term Owner arrived from a different tradition, which included HSMs and HSM appliances that could house multiple, virtual HSMs and might be used in non-PKCS environments.

The Owner is equivalent to Crypto Officer. The Owner is created when the Partition is created by the lunacm "partition create", and a 16-character challenge secret is generated by the PED.

The User is equivalent to Crypto User. The User is created by separate lunacm "partition createChallenge" command. A second, different 16-character challenge secret is generated by the PED.] of the Partition is the holder of the black PED Key for the Partition, and is the person who performs any Partition maintenance tasks (other than creating/destroying the Partition, which is done by the SO).

To allow your Client application(s) to work with the HSM means that they work with the Partition (the SO space is not normally used for operational and cryptographic purposes). Therefore, they must have an authentication secret to use when calling the HSM Partition to perform crypto operations. The basic authentication is the black PED Key, with which you login or Activate (cached login). If you have set up only the basic Partition arrangement, then the login or activation with the black PED Key is the only authentication that either the administrative User or the Client application needs.

You also have the option to impose an **additional level of security** by creating a text-based secret that the Client application can present (use the lunacm `partition createchallenge` command). The Luna PED generates that secret, and shows it on the PED screen, one time. You record it (preferably by typing into a text editor - handwritten text is easy to confuse...). That secret is then given to your Client application software when you configure that software to work with the HSM. Thereafter, the Client application presents the challenge secret whenever required (when a new session is opened and the Client app logs in).

However, because that challenge secret (also called Partition secret) has been imposed, it is also required when the Partition User/Owner wishes to use lunacm and run partition commands. The partition activate command caches only the black PED Key data, not the text challenge secret.

Finally, you can create a limited user, called the Crypto User (in other contexts, the regular Partition User / Owner might be called the Crypto Officer, to pair with this Crypto User designation). The Crypto User has a different Partition secret (the text challenge secret), and is able to use Partition objects, but not to manage/manipulate them (create, destroy, modify). The usual scenario would be to setup your HSM as SO, and to populate your HSM Partition with the required secrets, keys, certificates, while logged in as User/Owner, then finally to give out the Crypto User secret for use by the Client application, so that the Client app could use the existing Partition objects, but not modify them.

Password-authenticated HSMs

For Password Authenticated HSMs, the situation is simpler. There is no authentication hardware (no PED, no PED Keys). The Partition authentication is the Partition Password, and there is no separate challenge secret.

CHAPTER 8

PED Authentication

This chapter describes PED-based HSM authentication. It contains the following sections:

Compare Password and PED Authentication

	Password-authenticated HSM	PED-authenticated HSM
Ability to restrict access to cryptographic keys	• knowledge of Partition Password is sufficient • for backup/restore, knowledge of partition domain password is sufficient	• ownership of the black PED Key is mandatory • for backup/restore, ownership of both black and red PED Keys is necessary • the Crypto User role is available to restrict access to usage of keys, with no key management • option to associate a PED PIN (something-you-know) with any PED Key (something you have), imposing a two-factor authentication requirement on any role
Dual Control	• not available	• Mof N (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	• linked to password knowledge, only	• linked to partition password knowledge, • linked to black PED Key(s) ownership

	Password-authenticated HSM	PED-authenticated HSM
Role-based Access Control (RBAC) - ability to confer the least privileges necessary to perform a role	roles limited to: • Auditor • HSM Admin (SO) • Partition Owner	available roles: • Auditor • HSM Admin (Security Officer) • Domain (Cloning / Token-Backup) • Secure Recovery • Remote PED • Partition Owner (or Crypto Officer) • Crypto User (usage of keys only, no key management) for all roles, two-factor authentication (selectable option) and MofN (selectable option)
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

About the Luna PED

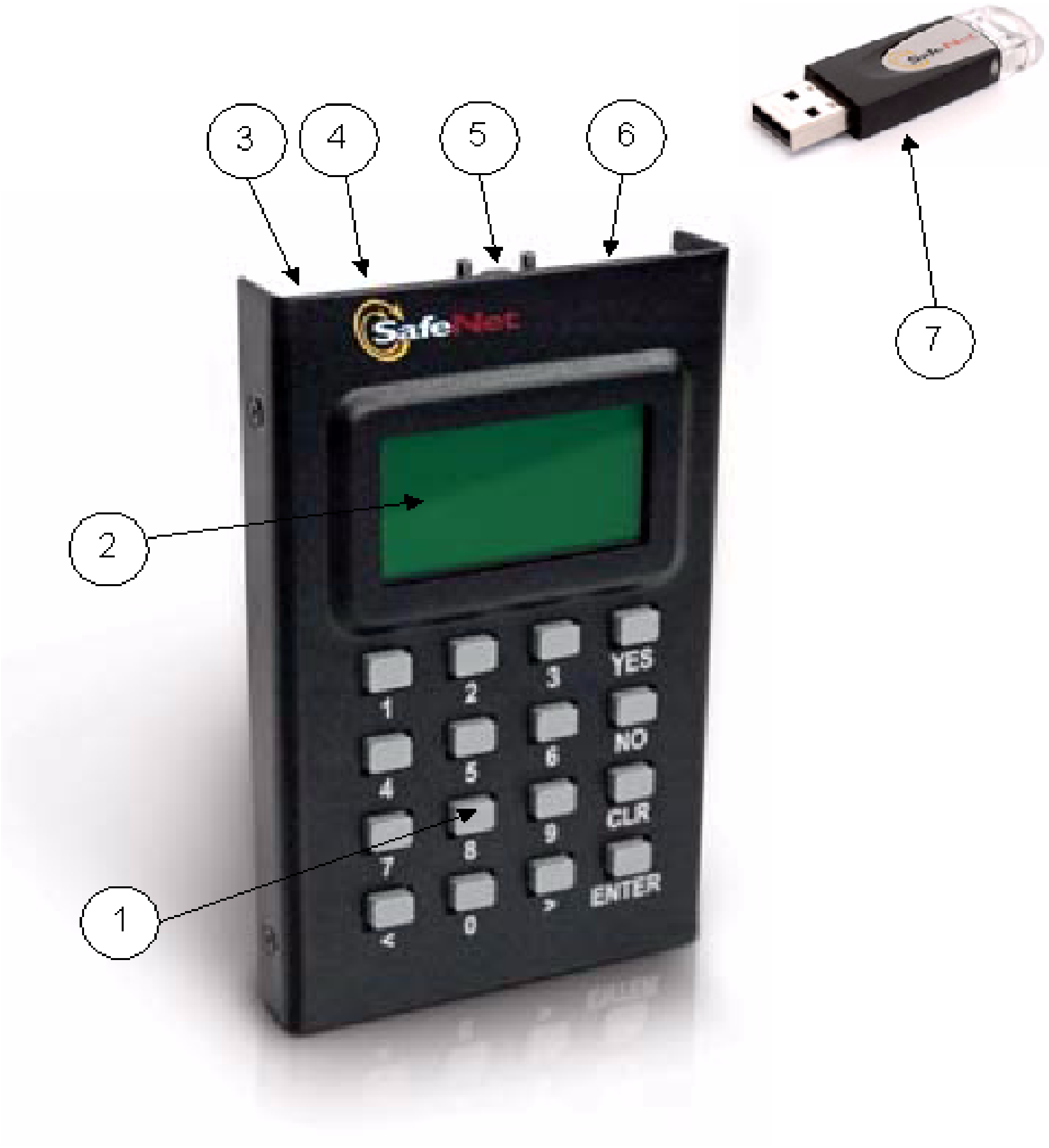
Luna PED is a PIN Entry Device, where PIN stands for Personal Identification Number. The PED works in conjunction with HSMs and backup tokens from SafeNet. It provides PIN entry to SafeNet HSMs and to backup tokens via a secure data port, as part of FIPS 140-2 level 3 security (the Trusted Path). PED 2.x is the current generation. A migration path is available if you have the legacy Luna PED 1.x.

The PED is shipped separately from your HSM product, because one PED can be used with any Trusted Path HSM. A PED with firmware version of 2.0 or later is also RoHS-compliant. The version is displayed on the PED display panel, each time the PED is powered on.

As well, you require a set of at least three PED Keys. For PED 2.0 and later, the PED Keys are in the form of hardware identification tokens, SafeNet iKey model 1000 (RoHS-compliant) or possibly other SafeNet iKey models, to be introduced at a later date. For most applications, you would want an additional set to make duplicates for backup purposes (and, optionally, several more PED Keys, if you intend to use the M of N authentication option with a SafeNet HSM product that supports M of N).

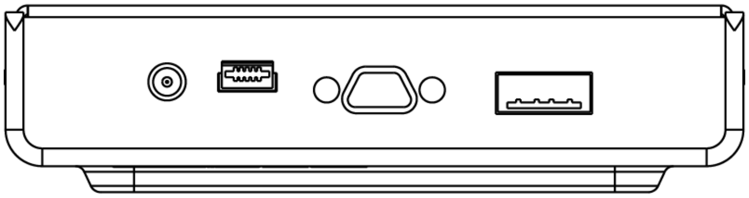
PED Features

The figure below shows a front view of the PED, with some important features indicated.



.

1. On the lower front face is the keypad for command and data entry.
2. On the upper front face is the 8-line liquid crystal display (LCD).



3. At the top on the far left is a DC power-adapter connector (not used when PED is connected directly to an HSM - local PED).

4. At the top, second from the left is a USB mini-B connector, reserved for file transfer to/from the PED.
5. At the top in the middle is a micro-D subminiature (MDSM) connector for the cable to the HSM (data and power).
6. At the top, on the far right, is the USB A-type connector for iKey-style PED Keys.
7. Also shown is an iKey PED Key, for insertion in the PED Key connector, and described in these pages.

The visible difference between the standard PED II (shown) and the Remote Capable PED 2 is the addition of "Remote Capable" on the back-panel label.

About PED Keys

A PED Key is an electrically-programmed device, with USB interface, embedded in a molded plastic body for ease of handling. Specifically, a PED Key is a SafeNet iKey authentication device model 1000 (must be firmware version 1.1 or later - the PED checks the firmware version of a presented iKey, and displays an error message if the version is too old) with FIPS configuration. In conjunction with PED 2 or PED 2 Remote, a PED Key can be electronically imprinted with identifying information, which it retains until deliberately changed.

A PED Key holds a generated secret that might unlock one or more HSMs. That secret is created by initializing the first HSM. The secret can then be copied (using PED 2.x) to other PED Keys, for purposes of backup, or to allow more than one person to have access to HSMs that are protected by that particular secret. The secret can also be copied to other HSMs (when those HSMs are initialized), so that one HSM secret is able to unlock multiple HSMs.

The HSM-related secret might be the access control for one or more HSMs, the access control for Partitions within HSMs, or the Domain key that permits secure moving/copying/sharing of secrets among HSMs that share a domain.

The PED comes in two versions:

- the standard PED 2 is designed for local connection, only, to a SafeNet HSM
- the Remote PED 2 has all the function of the standard PED 2 and can also be used remotely from an HSM, when used with PEDServer.exe workstation software.

Why do you need PED Keys?

The PED and PED Keys are the only means of authenticating, and permitting access to the administrative interface of the PED-authenticated HSM, and are the first part of the two-part Client authentication of the FIPS 140-2 level 3 (FIPS is the Federal Information Processing Standards of the United States government's National Institute of Standards and Technology -- FIPS 140-2 is an internationally recognized standard regarding security requirements for cryptographic modules, and level 3 is its second-highest level of security features/assurance) compliant SafeNet HSM with Trusted Path Authentication.

The use of PED and PED Keys prevents key-logging exploits on the host HSM, because the authentication information is delivered directly from the hand-held PED into the HSM via the independent, trusted-path interface. You do not type the authentication information at a computer keyboard, and the authentication information does not pass through the internals of the computer, where it could possibly be intercepted by spy software.

The PED does not hold the HSM authentication secrets. The PED facilitates the creation and communication of those secrets, but the secrets themselves reside on the portable PED Keys. This means that an imprinted PED Key can be used only with HSMs that share the particular secret, but PEDs are interchangeable(at least, within compatible versions - you can replace any PED 2.x with any other [unless otherwise indicated], but you cannot use a PED 1.x where a 2.x version is needed, or vice-versa) .

Types of PED Key

The current-model PED uses iKey USB-fob type PED Keys of no particular color (the standard issue is black) for all functions. You can visually differentiate your PED Keys by attaching tags or labels. A set of sticky labels in appropriate colors (see below) is supplied with your PED Keys.

The roles and uses of the PED Keys employed with SafeNet HSMs and the PED are as follows:

SO

Security Officer (SO)'s(also sometimes called HSM Admin) PED Key. The first actions with a new SafeNet HSM involve creating an SO PIN and imprinting an SO PED Key. The SO identity is used for further administrative actions on the HSMs, such as creating HSM Partition Users and changing passwords, backing up HSM objects, controlling HSM Policy settings. A PED PIN (an additional, optional password typed on the PED touchpad) can be added. SO PED Keys can be **duplicated**[1] for backup, and can be shared among HSMs by imprinting subsequent HSMs with an SO PIN already on a PED Key. See "Shared or Group PED Keys" on page 90.

Partition User

HSM Partition User key. This PED Key is required to login as HSM Partition Owner or Crypto Officer. Needed for Partition maintenance, creation and destruction of key objects, etc. Needed for the local portion of the login that permits remote Client (or Crypto User) access to the Partition. A PED PIN (an additional, optional password typed on the PED touchpad) can be added. Black User PED Keys can be **duplicated**[2] for backup, and can be shared among HSM Partitions using the "Group PED Key" option.

Domain

Key Cloning Vector (KCV) or Domain ID key. This PED Key carries the **domain**[3] identifier for any group of HSMs for which key-cloning/backup is to be used. The red PED Key is created/imprinted upon HSM initialization. Another (or could reuse the same domain) is created/imprinted with each HSM Partition. A cloning domain key carries the domain (via PED) to other HSMs or HSM partitions which are to be initialized with the same domain, thus permitting backup and restore among (only) those containers and tokens. The red Domain PED Key receives a domain identifier the first time it is used, at which time a random domain is generated by the HSM and sent to both the red Domain key and the current HSM Partition. Once imprinted, that domain identifier is intended to be permanent on the red Domain PED Key – and on any HSM Partitions or tokens that share its domain. Any future operations with that red Domain PED Key should simply copy that domain onto future HSM Partitions or backup tokens (via PED) so that they may participate in backup and restore

[1] a PED Key can be copied so that two or more PED Keys contain the same secret - this is useful and necessary in order to have backups of each of your PED Keys, and for other operational purposes, but you must maintain rigorous control of all duplicates to prevent unauthorized persons from accessing your HSM(s), and tracking of the "paper trail" of possession.

[2] a PED Key can be copied so that two or more PED Keys contain the same secret - this is useful and necessary in order to have backups of each of your PED Keys, and for other operational purposes, but you must maintain rigorous control of all duplicates to prevent unauthorized persons from accessing your HSM(s), and tracking of the "paper trail" of possession.

[3] (Also referred to as KCV – Key Cloning Vector) A domain is a shared identifier, common to a group of Luna cryptographic modules, with access controlled by a red PED Key (for Trusted Path Authentication) or by a domain string (for Password Authentication). Cloning (secure duplication) of token objects is possible among tokens/HSMs that share a particular domain. Cloning is not possible across different domains, and is not possible where the tokens lack a domain. A domain must be declared and imprinted at the time a token is initialized.

operations (see "Domain PED Keys" on page 92 for a more detailed explanation). Red PED Keys can be **duplicated**[1] for backup or multiple copies of the key.

The red PED Key can be considered the most important PED Key to protect from access by unauthorized persons. An unauthorized person who is able to learn the Luna SA appliance admin password, could see and manipulate objects on a logged-in or activated partition, but would be able to copy those objects to another HSM only if he had possession of the partition domain secret. Without the proper red PED Key, an attacker cannot copy/clone HSM partition contents to other HSMs.

Remote PED

This PED Key is required when you need to perform PED operations at a distance. The orange RPK carries the Remote PED Vector (RPV) and allows a Luna PED connected to a properly configured computer to substitute for a PED connected directly to the Luna appliance/HSM, when that local connection is not convenient.

The RPV is created/imprinted by a Luna HSM with a suitable PED connected (version 2.4.0 or later, having the Remote PED feature installed). A Remote PED can be connected to the USB port of a networked computer that has the PED driver installed and is running the PEDserver.exe program. A Luna HSM (that has been initialized with a Remote PED vector) can initiate a secure connection to the Remote PED Server computer, and that connection can be validated by an orange Remote PED Key that carries the same vector as the Luna HSM. For the duration of that session, HSM commands can be run at that appliance with all the needed PED Keys (SO, User, Domain, even SRK) being supplied via the PED connected to the computer. There is no need to be present at the remotely located Luna appliance/HSM with PED Keys and PED. Orange PED Keys can be **duplicated**[2] for backup or multiple copies of the key.

Secure Recovery

The purple Secure Recovery Key contains the external split of the SRV (secure recovery vector), to recreate the HSM's master key with which all HSM contents are encrypted. The master key is destroyed whenever a tamper event occurs, or when the HSM is deliberately set to Secure Transport Mode. For Secure Transport Mode, the purple PED Key is then shipped via a separate channel from the HSM shipment so that no attacker could obtain access to both the HSM and the SRV while they are in transit. Upon receipt, the administrator brings both the HSM and the purple key together, and invokes the "hsm srk recover" command. This brings the internal (in the HSM) and external (on the purple SRK) components of the SRV together and recreates the HSM master key, allowing the HSM to be used. Purple PED Keys can be **duplicated**[3] for backup or multiple copies of the key.

[1] a PED Key can be copied so that two or more PED Keys contain the same secret - this is useful and necessary in order to have backups of each of your PED Keys, and for other operational purposes, but you must maintain rigorous control of all duplicates to prevent unauthorized persons from accessing your HSM(s), and tracking of the "paper trail" of possession.

[2] a PED Key can be copied so that two or more PED Keys contain the same secret - this is useful and necessary in order to have backups of each of your PED Keys, and for other operational purposes, but you must maintain rigorous control of all duplicates to prevent unauthorized persons from accessing your HSM(s), and tracking of the "paper trail" of possession.

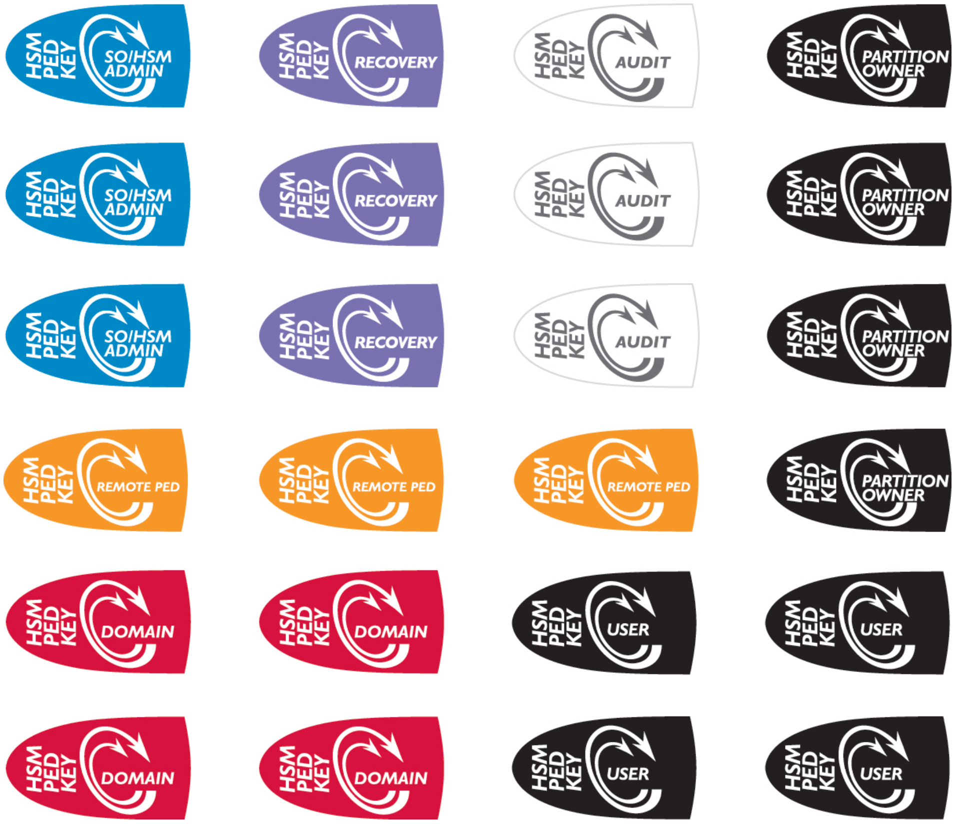
[3] a PED Key can be copied so that two or more PED Keys contain the same secret - this is useful and necessary in order to have backups of each of your PED Keys, and for other operational purposes, but you must maintain rigorous control of all duplicates to prevent unauthorized persons from accessing your HSM(s), and tracking of the "paper trail" of possession.

Audit

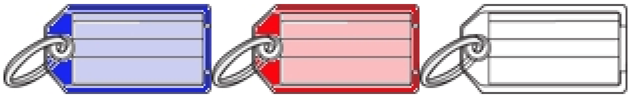
Audit is an HSM role that takes care of audit logging, under independent control. The audit role is initialized and imprints a white PED Key, without need for the SO or other role. The Auditor configures and maintains the audit logging feature, determining what HSM activity is logged, as well as other logging parameters, such as rollover period, etc. The purpose of the separate Audit role is to satisfy certain security requirements, while ensuring that no one else - including the HSM SO - can modify the logs or hide any actions on the HSM. The Audit role is optional until initialized. For Luna G5 and Luna PCI-E see the audit commands in lunacm:>. For Luna SA, there is a separate appliance login role (audit) that has access to its own lunash:> commands, in addition to a limited set of view-only commands for the HSM. The SO (a.k.a. HSM Admin) and others who log into the appliance as "admin" or as other named roles, do not have access to the lunash:> audit commands.

What is a Set of PED Keys?

A nominal set of PED Keys, as purchased with a SafeNet HSM with PED (Trusted Path) Authentication, consists of ten black USB-token PED Keys, along with colored stickers to identify them (several each of blue, red, black, orange, white, and purple), which allows some spares or backups. The PED Keys are completely interchangeable before they are imprinted by your action. The PED Keys are imprinted by the PED during HSM initialization and Partition creation, so at a minimum you would have one each of :



The stickers (above) are just visual labels to attach to your PED Keys. They are provided for your convenience, and you can use them, or not, at your discretion.



We recommend that you use some system of visually identifying the role of each PED Key once it is imprinted. Ordinary key-chain tags are handy and can be acquired anywhere; they provide room for written information that is important to you, and they do not interfere with the operation of the PED Keys.

We strongly suggest that you use our supplied self-stick PED Key labels, or that you otherwise maintain the color associations that are referenced throughout the documentation and also in the HSM utilities and the PED's own dialogs.

- Security Officer (SO) key (or might also be called HSM Admin) - blue
- domain key - red
- User key (or partition Owner key)- black
- Remote PED - orange
- Secure Recovery (SRK) - purple
- Audit role - white

The others are spares for each role. The SO, Domain, and User roles are the minimum that you need to operate the HSM.

For purposes of backup redundancy, you would normally have at least a second full set for keeping in safe storage, once they have been imprinted. Imprinting takes place when an HSM is initialized[1]. Initialization is also an opportunity to make more duplicates of any PED Key, if you require them. Imprinting of Partition PED Keys takes place when an HSM Partition is created[2]. Again, Partition backup is an opportunity to create more duplicate black PED Keys, or to cause a newly-created Partition to share an authentication secret that is already used on other HSMs' Partitions.

You will also require additional PED Keys if you decide to use the M of N security feature.

Physical Identification of PED Keys

This section is a few suggestions for your handling of PED Keys. Naturally you should be guided primarily by your organization's security policies.

As indicated above, you might wish to physically mark your PED Keys, in order to help keep track of them. Colored, blank tags are suggested, in addition to the provided stickers, though you could use any identifier that does not interfere with the operation of the PED Key. At a minimum, in an operational environment, you should have at least one working set and one full backup set, and a way to tell them apart.

If multiple personnel will need access to the HSM, you might provide duplicates of some or all PED Keys that are associated with a particular HSM. It would be helpful to number them, or to write the name or title of the person who will hold each duplicate, to ease tracking. Your organization's security policy might have requirements in that regard.

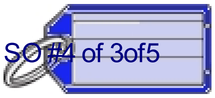
If you have multiple HSMs or groups of HSMs in your organization, a thoughtful labeling convention can ease the task of tracking and differentiating the various PED Keys and key-holder personnel.

[1] (a backup token is initialized/re-initialized whenever a backup is performed onto it)

[2] (on a SafeNet HSM it is always possible to create at least one Partition -- more may be possible, depending upon the configuration that you initially purchased, or upon licensing/capability update packages that you might later choose to purchase and apply)

If you invoke the optional M of N security feature (see the “Using M of N” page in this Help, you could have multiple sets of several PED Keys (containing the secret splits for SO or for the Partition Owner) that might require unique visible identification. Possibly one person might be the designated holder of M of N secret shares belonging to more than one HSM in your company. If that person is carrying several PED Keys, it would be convenient to see, at a glance, which PED Key belonged to which M of N set so as to avoid making accidental bad login attempts due to mix-ups of PED Keys.

For example, if each department in your company had a SafeNet HSM, and you were using M of N feature, your key tags might be labeled something like:

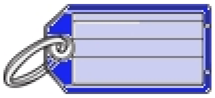


Accounts Receivable

SO #4 of 3of5

So this would be Security Officer (SO) key-share number 4, of a 5-key M of N set that requires at least three key-holders to be present to unlock the administration functions of that HSM in the Accounts Receivable department. You might prefer to not mention the "N" quantity, so that an attacker would not know how many more he/she needed to acquire.

Alternatively, you might use something obscure like:



AR4

which could be a code representing a more descriptive entry that you would keep in a log book or in a database. Either way, by looking at the tag you can quickly find out which of various PED Keys you are currently holding.

Obviously, these are just basic suggestions, and you can use any identifying scheme that works for you.

Using PED Keys

This is described in detail at "How to Use a Luna PED" on page 71, and in the Configuration and setup section of this Help.

Briefly, when you perform an HSM operation that requires a PED Key, you should already have the PED connected to the HSM or appliance.

When the command is issued, the system tells you when to look to the PED.

The PED prompts you when to insert various PED Keys, appropriate to the task. When prompted, insert the indicated PED Key into the connector at the top of the PED, immediately to the right of the PED cable connection, then respond to further instructions on the PED display, until control is returned to the administrative command-line.

What is a PED PIN?

For three-factor authentication, a PED PIN is "something you know", and is associated with "something you have", the PED Key (this is termed "three-factor" because you must:

- login to the password-protected [1st factor] admin session before you can invoke the HSM SO or User,
- provide a physical PED Key [2nd factor] and
- input the optional PED PIN [3rd factor]).

A PED PIN is an optional additional authentication layer(It is optional only for the first PED Key imprinted at initialization time - if you choose to have some duplicates made of that PED Key, then they all get the flag for PED PIN [or no PED PIN if that's what you chose] that you gave for the first key.) for any of:

- the HSM Admin or SO (blue PED Key) or,

- the Partition Owner or Crypto Officer (black PED Key)
- the cloning Domain (red PED Key)
- the Remote PED Key (orange PED Key)
- the Secure Recovery Key (purple PED Key)
- the Audit key (white PED Key).

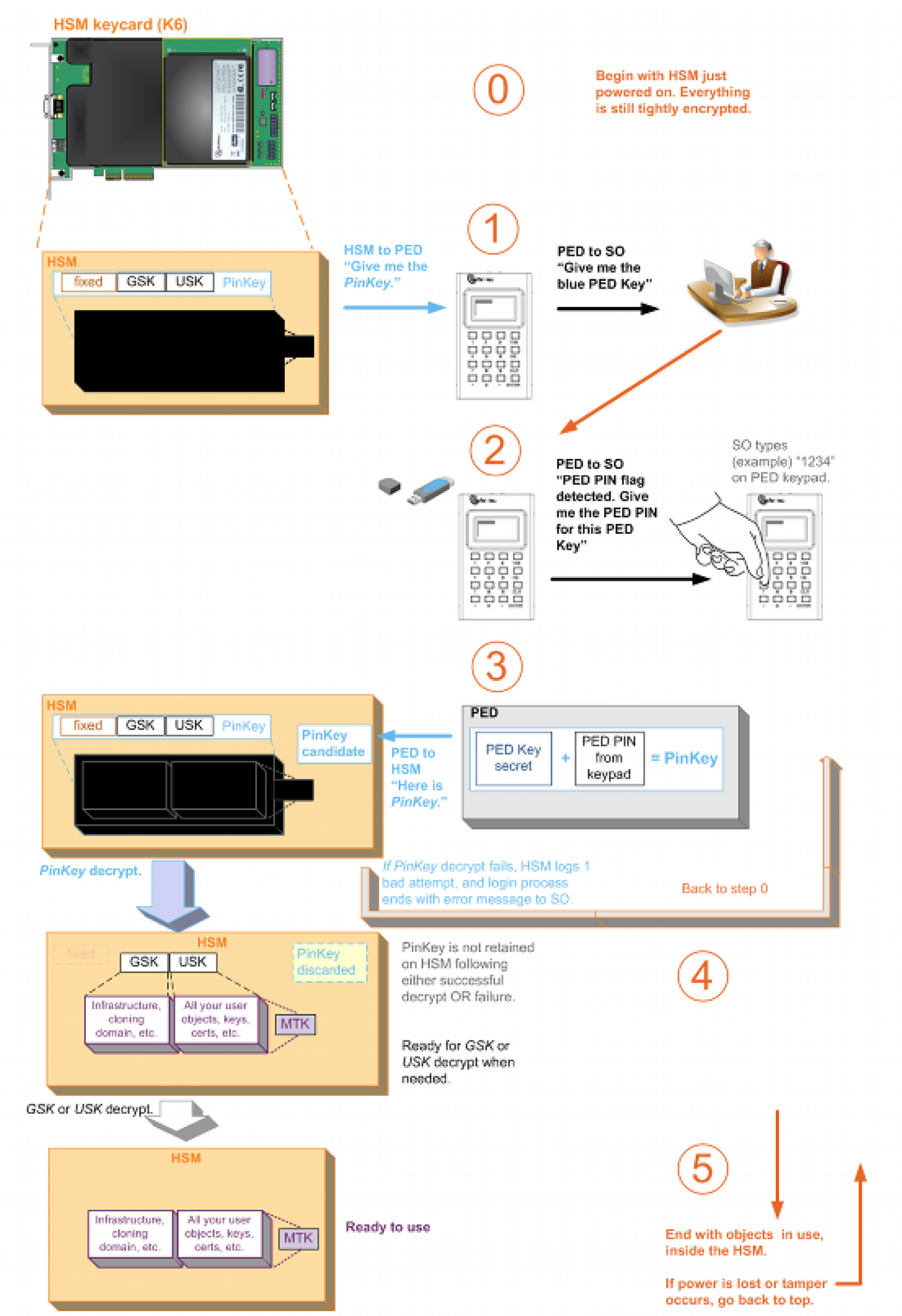
The secret that unlocks the HSM is the PinKey.
In Password authenticated HSMs, the PinKey is the text password that you type at a keyboard.
In PED authenticated HSMs, the PinKey is the secret that the HSM receives from the PED when the HSM calls for authentication.

But what is it?

A PED PIN is a sequence of digits that you type in, at the PED keypad, which is combined with the secret stored on the key, and the resulting combined PinKey is sent to the HSM. The combined secret-and-PED-PIN is what the HSM recognizes as its unlocking secret.

Here is a diagram that might clarify the concept. Click the small picture to open a larger, readable version in a new tab or window.

HSM – PED interaction, one PED PIN


HSM keycard (K6)
0
Begin with HSM just powered on. Everything is still tightly encrypted.
HSM
fixed
GSK
USK
PinKey
1
HSM to PED "Give me the PinKey."
PED to SO "Give me the blue PED Key"
2
PED to SO "PED PIN flag detected. Give me the PED PIN for this PED Key"
SO types (example) "1234" on PED keypad.
3
HSM
fixed
GSK
USK
PinKey
PinKey candidate
PED to HSM "Here is PinKey."
PED
PED Key secret
+
PED PIN from keypad
= PinKey
PinKey decrypt.
If PinKey decrypt fails, HSM logs 1 bad attempt, and login process ends with error message to SO.
Back to step 0
HSM
fixed
GSK
USK
PinKey discarded
Infrastructure, cloning domain, etc.
All your user objects, keys, certs, etc.
MTK
PinKey is not retained on HSM following either successful decrypt OR failure.
4
Ready for GSK or USK decrypt when needed.
GSK or USK decrypt.
HSM
Infrastructure, cloning domain, etc.
All your user objects, keys, certs, etc.
MTK
Ready to use
5
End with objects in use, inside the HSM.
If power is lost or tamper occurs, go back to top.

If, for example, you are initializing an HSM and not re-using any existing secret on the PED Key that you present (or it's a blank key), then during the process, the Luna PED prompts you to provide a PED PIN. (see below)

How to invoke/require a PED PIN with an HSM

At the Luna PED prompt:

Enter a new PED PIN

If you want a PED PIN:

- enter 4 to 48 digits via the Luna PED keypad and press [Enter] (you are prompted to enter the PED PIN again, to confirm)
  Note: do not use zero for the first digit

  (When the leading digit is zero, the PED ignores any digits following the exact PED PIN. Thus an attacker attempting to guess the PED PIN must get the first digits correct, but does not need to know the exact length of the PED PIN. If the PED PIN is started with any digit other than zero, extra digits are detected as an incorrect attempt. This is not considered a real vulnerability since any attacker
  a) must have physical possession of the PED KEY,
  b) must have physical access to the HSM and PED, and
  c) gets only three tries to guess correctly, before the HSM is zeroized.
  However, since we noticed it, we thought we should mention the slightly different function when the first PED PIN digit is zero.)
- the PED PIN must be the same across multiple HSMs
- Luna PED combines your PED PIN with the random PIN from the (blue or black) PED Key and presents that combination to the token as the authentication for HSM Admin or the Partition Owner (or Crypto Officer) respectively
- PED PIN digits are not echoed to the PED screen; instead, whatever you type is masked by asterisk (*) characters.

If you don't want a PED PIN:

- just press [Enter] on the Luna PED keypad (signifying 0 digits); you are prompted again, to confirm.

The PinKey is the secret on the PED Key, combined with the PED PIN. The PED PIN is not recorded - it is a transformation that you perform on the PED Key secret to convert it into the PinKey.Therefore, the PED PIN is separate and distinct from the HSM SO authentication secret (or the User/Owner/Crypto Officer authentication, etc.) contained on the PED Key. It is optional to create a PED PIN (as an extra layer of authentication security) when you initialize an HSM, but once a PED PIN is invoked, it is then required every time you authenticate to the HSM. That is, if you opt to not create a PED PIN at initialization time (or Partition creation time for the black PED Key), then you never use PED PINs, but if you do create a PED PIN at initialization time, then you are "stuck" with the requirement until the next time you wipe the contents (zeroize) and re-initialize. The point to make is that the PED PIN option is there if your policy and situation require the additional security, but you don't need to invoke the extra layer if you don't require it.

The choice to invoke PED PIN for a particular PED Key function [blue SO key, black User/Owner key, red Cloning Domain key, orange Remote PED key, white Audit key, or purple Secure Recovery key] is independent of the other types of PED Key.

For example, if (at initialization time) you decide to have a PED PIN for the blue (SO) PED Key, then that PED PIN is thereafter required when you use blue PED Keys with that HSM(until you initialize again) , but you do not need to use PED PINs with the black and red PED Keys if you don't wish to do so. Similarly, you might choose to invoke PED PIN for the red PED Key, but not for the blue or black PED Keys.

Here are possible combinations if you have two HSMs H1 and H2, and any of several initialization-time choices regarding PED PIN. What is important to unlock the HSM is the secret that is imprinted on the HSM, so in the following table we will call that secret H1SO or H2SO. We will call the secret contained on the PED Key K1SO or K2SO.

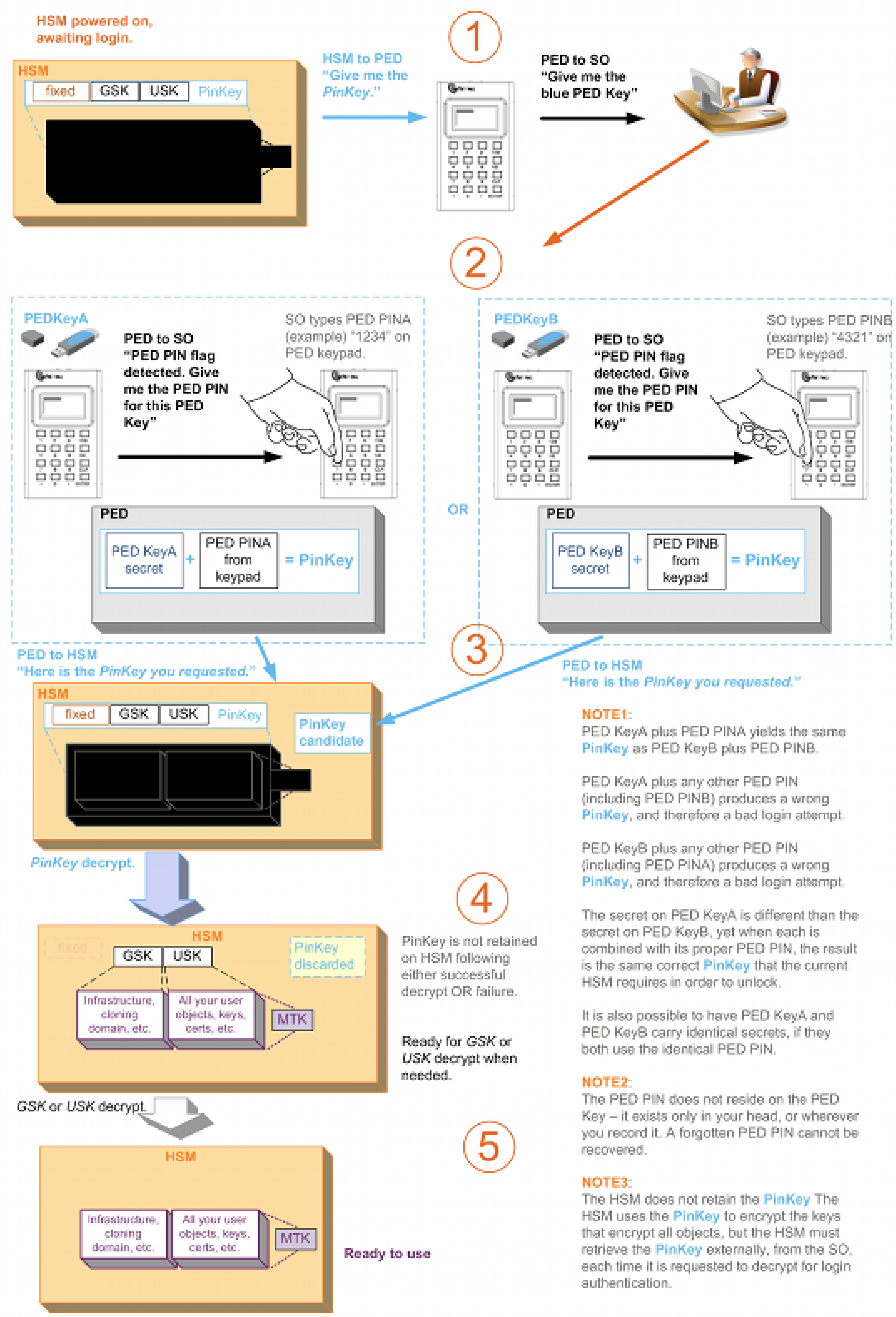
Configuration	SO Authent Secret on HSM	What You Need to Unlock HSM	PED Keys Interchangeable?
Different blue PED Key Pinkeys H1SO and H2SO K1SO does not equal K2SO	H1SO not same as H2SO	The correct PED Key for the current HSM	No

Two identical blue PED Keys, no PED PINs, so PED Key secret is the PinKey secret, which is the same on both
K1SO = K2SO and
H1SO = H2SO
	H1SO secret identical to H2SO	Either PED Key; both are correct for either HSM	Yes

Two identical blue PED Key(s), same PED PINs so PED Key secret is the same on both (K1SO = K2SO) and therefore PinKey secret is the same for both, to yield H1SO = H2SO	H1SO secret identical to H2SO	Either PED Key plus the one PED PIN; both are correct for either HSM	Yes
Two blue PED Key(s), different PED PINs for both HSMs, but PED Key secrets are also different (K1SO does not equal K2SO) such that PED Key1 plus PED PIN1 together generate the same PinKey secret as PED Key2 plus PED PIN2 - H1SO = H2SO	H1SO secret identical to H2SO	Either PED Key plus the correct PED PIN for just that PED Key; both are correct for either HSM BUT Either PED Key with the PED PIN for the OTHER PED Key is a bad login attempt	Yes, but the PED PINs are not.

Here is a drawing of HSM PED authentication with two PED PINs. Click the small picture to open a larger, readable version.

HSM – PED interaction, duplicate PED Keys, two different PED PINs


HSM powered on, awaiting login.
HSM
fixed
GSK
USK
PinKey
1
HSM to PED "Give me the PinKey."
PED to SO "Give me the blue PED Key"
2
PEDKeyA
PED to SO "PED PIN flag detected. Give me the PED PIN for this PED Key"
SO types PED PINA (example) "1234" on PED keypad.
PED
PED KeyA secret
+
PED PINA from keypad
= PinKey
OR
PEDKeyB
PED to SO "PED PIN flag detected. Give me the PED PIN for this PED Key"
SO types PED PINB (example) "4321" on PED keypad.
PED
PED KeyB secret
+
PED PINB from keypad
= PinKey
3
PED to HSM "Here is the PinKey you requested."
PED to HSM "Here is the PinKey you requested."
HSM
fixed
GSK
USK
PinKey
PinKey candidate
PinKey decrypt.
4
HSM
fixed
GSK
USK
PinKey discarded
Infrastructure, cloning domain, etc.
All your user objects, keys, certs, etc.
MTK
GSK or USK decrypt.
5
HSM
Infrastructure, cloning domain, etc.
All your user objects, keys, certs, etc.
MTK
Ready to use

PinKey is not retained on HSM following either successful decrypt OR failure.

Ready for *GSK* or *USK* decrypt when needed.

NOTE1:
PED KeyA plus PED PINA yields the same PinKey as PED KeyB plus PED PINB.

PED KeyA plus any other PED PIN (including PED PINB) produces a wrong PinKey, and therefore a bad login attempt.

PED KeyB plus any other PED PIN (including PED PINA) produces a wrong PinKey, and therefore a bad login attempt.

The secret on PED KeyA is different than the secret on PED KeyB, yet when each is combined with its proper PED PIN, the result is the same correct PinKey that the current HSM requires in order to unlock.

It is also possible to have PED KeyA and PED KeyB carry identical secrets, if they both use the identical PED PIN.

NOTE2:
The PED PIN does not reside on the PED Key – it exists only in your head, or wherever you record it. A forgotten PED PIN cannot be recovered.

NOTE3:
The HSM does not retain the PinKey The HSM uses the PinKey to encrypt the keys that encrypt all objects, but the HSM must retrieve the PinKey externally, from the SO, each time it is requested to decrypt for login authentication.

Must I Use a PED PIN?

If a PED PIN has been set for a PED Key and an HSM, then you must always provide that PED PIN when using that key (or any duplicate of it) to login to that HSM. If you duplicate a PED Key, what you are duplicating is the secret that was originally imprinted on the PED Key, plus the state of a flag. The flag is an instruction to the PED to "prompt for a PED PIN"... or not.

If you choose, at initialization, not to invoke a PED PIN (that is, if you just press [Enter] without typing any digits on the keypad), then the flag is not set on the PED Key, and the secret on the PED Key matches exactly the secret in the HSM. Any duplicates that you make of the first PED Key will also have the flag unset. Whenever you use any of those PED Keys (original or duplicates) the PED checks for the state of the flag, finds it not set, and simply decrypts and sends the unmodified stored secret to the HSM, without prompting for PED PIN.

Should I Use a PED PIN?

That is up to you and your organization's security policy, but security procedures should never be more complicated than your requirements dictate.

Consider also if your security policy requires regular changes to passwords and other authentication. Your personnel would need to remember new PED PINs with each change cycle. If people are asked to remember too many passwords/PINs or asked to change them too often, they begin writing them down, which is itself a potential security issue.

What If I Change My Mind?

You can remove the requirement for a PED PIN by using the 'hsm changePw' command. A new secret is generated on the HSM, and is imprinted onto the PED Key (you are asked if you want to overwrite the existing data and you say YES). You are given the opportunity to add a PED PIN and you just press ENTER on the PED keypad to decline a PED PIN.

During the PED operation, you are given the opportunity to imprint additional keys with the new secret that doesn't include a PED PIN. You can use that opportunity to imprint additional new, blank PED Keys, or to overwrite PED Keys that are already imprinted with the old secret[1].

This action must be performed on all the PED Keys [duplicate PED Keys] associated with that HSM.
If you have a group of HSMs that share the same authentication secret (meaning they can all be unlocked by the same PED Keys [group PED Keys, see below]) then you must keep one unchanged PED Key until you have logged in and performed the 'hsm changePw' command on all the HSMs in that group.

Similarly, if you decide to increase the stringency of your security, you can use the 'hsm changePw' command to change the secret on your PED Keys and HSM(s) and at the same time, add PED PINs. Again, if you make such a change, consider doing it on all copies [duplicates] of the PED Key, and on all HSMs that shared the old PED Key authentication data.

Alternatively, you could leave some PED Keys with the old secret and leave some HSMs with that same secret. The result would be two groups of HSMs and associated PED Keys that could not be interchanged (for authentication). In other words, you could use that technique to split a group of HSMs.

Does that apply to the other PED Key colors?

Not all.

[1] [which is now invalid for the current HSM]

- It does apply to the black PED Key - use the lunacm command `partition changePw`. This change is non-destructive to the HSM partition or its contents.
- For the purple PED Key, you must generate a new SRK (lunacm command `srk keys resplit`). This requires that you have the old/current SRK to begin, and that you provide a different PED Key to receive the new Secure Recovery Vector. The PED does not allow you to overwrite the current purple PED Key. This change is non-destructive to the HSM or its contents.
- For the orange PED Key, you can use the lunacm command `ped vector init` to create a new Remote PED vector on the HSM and on the current orange PED Key, or you can import a different RPV from a different orange SRK and imprint that RPV onto the HSM in place of the current one. This change is non-destructive to the HSM or its contents.
- However, you **cannot** change an HSM Domain without a hard initialization of the HSM (destroys all contents), and you cannot change a partition Domain without deleting the current partition and creating a new partition, which deletes all contents of the current partition.

What is a Shared or Group PED Key?

Visit this topic for an additional, interesting concept that might be important to you when imprinting and using PED Keys:
See "Shared or Group PED Keys" on page 90.

What else do I need to know?

Here is a re-cap of what happens when you initialize.

The HSM, when told to initialize, turns over control to the PED, which immediately 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.

The secret (whether from the current HSM or from an inserted PED Key, previously imprinted by another HSM) is presented to the PED.

If you are using a new secret[1], the PED prompts for a PED PIN, and you provide either a string of digits via the keypad (a PED PIN), or no digits and just a press of "Enter" (no PED PIN).

If you are reusing an existing secret, then the PED takes that from the presented PED Key (including any PED PIN, which you must know and provide when prompted) and presents that to the HSM.

At this point, either the secret from the HSM is written to the PED Key, or the secret from the PED Key (possibly combined with a PED PIN is written to the HSM. If a PED PIN exists, then the secret on the PED Key is modified from the original by combination with the PED PIN, and that modified secret is imprinted upon the HSM - only the unmodified secret on the PED Key, combined with the PED PIN can reproduce the secret that the HSM expects.

The PED asks if you will be duplicating this key. Each duplicate can have a different PED PIN (or no PED PIN).

The same pattern applies for any of the secrets - SO (blue), User/Owner (black), Domain (red), RPK (orange), SRK (purple).

[1] [you answered "NO" to the "...reuse..." question]

Best Practice

When you initialize a PED-authenticated HSM (or create a partition, or perform any action that imprints a PED Key), and you choose to associate a PED PIN with the PED Key secret, you must ensure that the PED PIN will be remembered when it is needed. That normally means writing it down on paper or recording it electronically. This, of course represents a security risk. But it would equally be a security risk to not record the PED PIN and then be unable to remember it.

Before you tuck that yellow-sticky with the PED PIN into your safe, TRY it once, to verify that you did set the PED PIN that you think you set (or that you correctly recorded what you actually set).

In the case of a red key, that would mean you would need to attempt a cloning or backup/restore operation before storing your record of the PED PIN.

What is M of N?

M of N is the Luna HSM (the Trusted Path version [FIPS 140-2 level 3 compliant] that uses Luna PED and PED Keys for authentication) access-control feature that implements a split-secret (or split-knowledge) threshold scheme to divide and distribute the HSM authentication among multiple holders. Also sometimes called Multi-person Control or quorum-based authentication, this model is part of a layered security strategy, and ensures that no single person has sufficient authority to access certain functions or operations.

The split secret can be the HSM Security Officer authentication secret, or the Cloning Domain secret, or the User secret, any of which can be divided into N different parts and distributed among multiple holders. The threshold is the number M of splits out of the total N that must be recombined in order to reconstruct the complete secret and gain access to the HSM.

Setting Up M of N

M of N is decided at initialization time. The Luna PED prompts for the number M and the number N before the newly generated secret is imprinted.

The PED prompt asks:

```
M value? (1-16)
>0
```

Enter a number for M (the threshold value) on the PED keypad and press [ENT].

The PED then prompts:

```
N value? (M-16)
>0
```

Enter a number for N (the size of the full set, the total number of shares into which this secret is to be split).

The PED performs the split and sets the threshold, and then begins prompting you to insert blank PED Keys, to be imprinted.

If M and N are set to 1, then the secret is not split, and the entire secret is imprinted onto a single PED Key. In that case, M of N is effectively turned off, or not used.

Note: You can make additional copies of the single PED Key (at a later step in the initialization), but each key in this situation will have the complete authentication secret, therefore any one of them will be sufficient to authenticate to the HSM. In contrast, if you had declared M and N to be greater than one, then any key that you duplicated would have a copy of only a split (an incomplete portion) of the authentication secret. Two identical PED Keys from an MofN split cannot be used as separate portions to recreate a split authentication secret - therefore, you must be very careful when handling and labeling PED Keys if you have:

Note: a) invoked MofN (by setting M and N greater than 1) and

Note: b) created duplicates of any/all of the PED Keys containing those splits.

Note: In other words, if you invoke MofN and you also create duplicates, be very careful to keep the "original" and the backup PED Key sets identified and separate from each other.

If M and N are set to greater than 1, then M must always be less than, or equal to N. The maximum is 16 splits of the secret. If M=N, then every one of the splits must be recombined in order to reconstruct the complete secret. In that case, you must be very sure of the reliability of all key holders, since if any of them is sick, away on vacation, or otherwise not available, then access to the HSM is not possible.

The usual practice is to determine a reasonable number M that is sufficient for your security policies, and then allow a few more splits as "spares", the total number being your chosen N. This allows some operational flexibility, at the cost of validating additional trusted key-holder personnel.

Additional Considerations

The Luna Identity Server HSM and Luna PED offer additional, optional security features, all of which can be used together in any combination, if desired. You and your security policy must determine which features are necessary in your situation. Keep in mind that your handling procedures will need to adapt to the results of your choices.

Duplicate PED Keys

During initialization, Luna PED offers the option to create duplicates of imprinted PED Keys. Without M of N, you can make as many duplicates as you wish of a PED Key, or as many as you have blanks. With M of N invoked, you are guided by the Luna PED to submit a full set of N blanks for each duplicate. That is, if you desired more than one backup duplicate, you would be prompted to insert N blank keys in succession to complete one M of N backup set and then asked if you wish to imprint another. You must have a full set of blanks available to create each backup set.

If you run out of PED Keys before a set is complete, the system eventually times out, possibly leaving itself in an indeterminate state. Therefore, you would need to restart the initialization and either arrange to have enough blanks available, or choose to make fewer duplicates.

Unless your security policy forbids, you should carefully label all PED Keys. This will make them much easier to administer in situations such as personnel turnover, mandated password-change cycles, and so on. Physical labelling will allow you to distinguish which PED Key belongs with which person, which keys belong to primary or backup sets, and possibly other operational considerations. If you make a duplicate set, the same splits are generated, in the same order.

PED PINs

If your security policy demands three-factor authentication ("something you know" [the alpha-numeric admin password] as well as "something you have" [the PED Key], along with an additional "something you know" [the numeric PED PIN]), then you can use the PED PIN option. A PED PIN is a number that you type in at the PED keypad, and which becomes associated with the secret-split on the PED Key, masking that partial secret.

Note: If you prefer not to use PED PINs, just press [ENT] at the prompt, without typing any digits on the PED keypad.

You can apply any PED PIN number to a PED Key when that PED Key is being imprinted. Thus you could use the same PED PIN on every Key, or a different one on every Key. You could use matching PED PINs on the matching Key from each of two (or more) sets, or make them all different. Each option affects the convenience or the effectiveness of the additional level of security.

The last sentence in the previous section said: If you make a duplicate set, the same splits are generated, in the same order. That is still true, when you invoke PED PIN security, but the PED PIN portion of the data on the PED Key is not necessarily the same (Every time you are given an opportunity to impose a PED PIN onto a PED Key, you are free to give any number [either none, or a number from 4 to 48 digits] -- there is never a constraint to match any PED PIN that you used on other PED Keys, such as duplicates or M of N splits. The system dictates no requirements of that nature -- every time you are given the opportunity to impose a PED PIN, you have a new opportunity to decide whether or not to have a PED PIN and, if so, the digits that compose that PED PIN. Of course, when you later attempt to authenticate with a PED Key, you must supply the exact PED PIN that was set for this particular PED Key [if one was].) unless you choose to input the same PED PIN digits on equivalent splits of the new set.

Use PED PINs in conjunction with M of N only if you really need both features and are prepared to deal with the resulting logistic requirements.

Using M of N

Once M of N is set, and initialization completed, the HSM and Luna PED take care of enforcing the feature. Whenever you attempt an action that requires login, the Luna PED begins prompting for M different PED Keys of that type. It continues until it has collected enough parts (splits) to reconstruct the original secret. It detects attempts to re-use a key during a single authentication attempt, and is not satisfied until it has gathered M different components of the split secret.

Only the splits from the original secret are valid - that is, you cannot substitute a key from a different set; Luna PED refuses to accept any individual key that is not a unique member of the current split secret. When it has collected M different splits from the correct set, it offers the reconstructed secret to the HSM and login procedes.

To help implement such a policy around a Luna HSM, the M of N Access to those functions or operations can require the presence and co-operation of multiple trusted persons, simultaneously. Such a strategy is sometimes called shared integrity or split-secret threshold. For example, if you are a Security Officer for an HSM, your authentication key alone would not be sufficient for you to administer that HSM - you need the cooperation of a specified number of other SOs, who must concur and authenticate when you do.

The Luna Identity Server M of N feature provides a means by which organizations employing cryptographic modules for sensitive operations can enforce multi-person shared authentication control over access to the cryptographic module. The feature is available in all Luna Identity Servers configured to use Trusted Path authentication – using the PIN Entry Device (PED) and PED Keys.

M of N involves a modification of the SO PIN, User PIN or Domain secrets. M of N causes the PIN or Domain secret to be divided and shared (or “split”) among several PED Keys of one type (“split-knowledge access control”). M of N is optional at initialization time.

M of N is offered as a choice by the Luna PED when PED Keys are being imprinted. N is the size of the pool of shares or splits. M is the number of shares (each on its own PED Key) that must be brought together to reconstruct the full authentication secret.

The PED prompt asks:

```
M value? (1-16)
>0
```

Enter a number for M (the threshold value) on the PED keypad and press [ENT].

The PED then prompts:

```
N value? (M-16)
>0
```

The PED performs the split and sets the threshold, and then begins prompting you to insert blank PED Keys, to be imprinted.

The same general pattern applies to the red Domain PED Keys and the black Partition/group User PED Keys. You can choose to impose the M of N requirement on any of SO (blue), Domain (red) or User (black), without requiring M of N for the others. Similarly, if two or more kinds of PED Keys have the M of N secret sharing imposed, the M and N values can be different for each set. That is, if you invoked M of N as 3 of 5 for SO, you could require (say) the Domain to be 2 of 3 and the User authentication to be 4 of 6, or whatever combination suited your policies and procedures.

M of N is ***not*** a splitting of the private signing key; it is splitting of the Luna Identity Server HSM's authentication secret.

Using the PED

PED (2.x) is required when you wish to authenticate to your HSM with PED (Trusted Path) Authentication.

The requirement for Trusted Path Authentication, as opposed to Password Authentication, is part of the specific model of HSM, as configured at the factory.

The PED does not contain any authentication information. PEDs are interchangeable (within the version range, Luna PED 1.x or PED 2.x) - it doesn't matter which local PED 2.x you use. The authentication information is contained on the PED Key, and PED is the device that provides the interface so that authentication data can pass between PED Key and HSM.

A locally-connected PED is powered by its connection to the HSM appliance. That connection - directly between the PED and the HSM card inside the appliance - bypasses your computer bus and the computer bus of the appliance. It is the only data path between the HSM and the PED and therefore is considered much more secure (trusted) than any authentication path that passes through the appliance's computer data paths. The Trusted Path cannot be monitored by any software (whether authorized by you or not) on your administrative or client computer. Also, because you use the PED Keypad to input the optional PED PIN password (to unlock the secret that, in turn, unlocks your HSM), nothing is typed on a computer keyboard. No virus, trojan, spyware, remote-session software or other method can be used to acquire those secrets, because they never pass through the normal computer data pathways, never reside in computer memory or on disk.

With HSM appliances normally tucked away in server farms, which are often run as "lights-off" facilities with the minimum possible human intervention, the PED cannot always be conveniently connected directly to the HSM. Instead, a callback server arrangement (Remote PED) uses a Luna PED connected to a separate computer, at a convenient location, to serve PED interactions over a network connection. The connection is strongly secured and, like the direct connection, prevents unauthorized persons from gaining access to the authentication data.

The only way for another person to discover a PED PIN password while you are inputting it is if you allow that person to watch while you use the PED keypad.

When Do I Need A PED?

You need to use the PED whenever you perform any action with the HSM that causes it to look for authentication (with some exceptions, see below). For example, using the Luna shell (lunash) you might login as Security Officer, login as User, or initialize the HSM. When the HSM receives such a command, it requests the appropriate data from the PED - or in the case of initialization, the HSM might send data to the PED.

Therefore, you should have the PED connected and in its ready state ("Awaiting command...") when you issue a command that invokes the PED. One MDSM connector attaches to the matching connector on the HSM or appliance, and the other MDSM (Micro-D Sub-Miniature) connector attaches to its matching connector on the top of the PED (tighten the connector screws if you intend to leave the PED connected; this makes the most reliable connection and provides strain relief to the cable-connector junction during handling of the device).

If you are using the Activation/autoActivation feature then, after authentication, the data is cached on the HSM, where it remains until you deactivate or you remove power to the HSM. In that case, once the authentication is performed, you can disconnect the PED and store it until the next time it is required.

If you are not using autoactivation, then authentication data is not cached and every time you or your client application needs access to the HSM, the HSM will look to the PED, which must remain connected.

What Do I Do?

As soon as it receives power from a to a powered appliance, the PED performs its startup and self-test routines and then goes to its normal operating mode, SCP mode, displaying the prompt "Awaiting command...". The PED is ready for use.

There are two things that you can do with the PED at this point:

- Wait for a prompt, which results when a program has caused the HSM to request authentication
- Perform standalone PED operations.

To perform prompted actions, just do what is asked on the PED screen. Normally the prompts are:

- Insert a PED Key
- Press "YES", "NO" or "ENTER" on the keypad

Insert and remove appropriate PED Keys, type passwords when requested, and so on. The particular sequence depends upon the operation that the HSM needs at the time, which in turn depends on the command-line administrative operations that you are performing (with lunacm, lunadiag, multitoken2, or other SafeNet utilities), or operations triggered by your applications.

In normal practice, you would perform initial configuration operations one time before placing the unit in service, then perform only monitoring and occasional maintenance thereafter. See the table below for a simple breakdown of the normal tasks and if/how the PED and PED Keys might apply.

Situation	Needs	Action with PED and PED keys
Setup/configuration	Appliance admin password, blue, red and black PED Keys and	You perform the HSM initialization, create Partitions,

Situation	Needs	Action with PED and PED keys
	PED. Network connection to the appliance from your administrative PC, and preferably also a local serial connection.	set up Remote PED, set up a redundant, load-sharing cluster with other SafeNet HSMs. This is the kind of chore you must perform before first putting the unit into "production", and then might never need to do again. The PED Keys are required at several stages, as well as the PED.
Occasional Maintenance of HSM	Appliance admin password, blue and black PED Keys, possibly the red if you need to initialize a new cluster member, and the PED. Network connection to the appliance.	Add and remove cluster members, modify number and assignment of Partitions/Groups, enable and disable... you might need some or all PED Keys for authentication, depending on the activity.
Client access	Client applications need their own authentication which, for PED-auth HSMs, is the challenge secret; no PED Key or PED required once the Partition is activated.	None. You would normally have activated/auto-activated the , and put the PED and PED Keys away in safe storage. They aren't needed in ongoing operation.
PED Key administration	A PED and whichever PED Keys you wish. You can connect to any SafeNet HSM that has the proper connector - this is to power the PED only. Alternatively, you can use the PED power supply kit provided with PED 2 (Remote Capable), and not need any HSM connection.	While you can perform some PED Key admin during HSM operations (mentioned elsewhere), you can also just power up the PED, go to Admin mode (instead of the default "Local PED" mode), and perform actions like creating duplicates of your existing, imprinted PED Keys. No HSM access is required. See the next section on this page (below) for more detail.

Standalone or local or off-line PED operations

You can perform some operations on PED Keys without going through the HSM.

To perform standalone operations

1. Press the "<" key to exit from SCP mode.
2. In Admin mode, select 1 PED Key or 7 Software Update. (The software update function is rarely used and requires that you be sent a PED software file from SafeNet, along with directions about how to use it. Therefore, we'll assume that you are selecting "1 PED Key", which brings the PED to PED Key mode.)
3. To perform an operation on a particular PED Key, insert that PED Key into the PED Key connector on top of the PED.
4. PED Key mode has an option "1" to login to that PED Key, which applies to models other than iKey 1000 PED Keys - just press "1" to get to the next menu, if you are using iKey 1000 PED Keys, which don't need login.

5. At the PED Key Mode menu you have options to Login (which you have just done, but the prompt is available in case you might wish to login to a different PED Key) , Logout, or Duplicate the PED Key. Only the "Duplicate" option is meaningful for your iKey 1000 PED Key. To **duplicate** the contents of the currently connected and PED Key to another (blank or re-used) PED Key, press "7" on the PED keypad.
6. Insert a blank target PED Key, or a non-blank whose data is no longer needed, and press ENTER.
7. If data already exists on the target PED Key, you are warned and required to press YES two times, to confirm that you really do wish to overwrite whatever is on the PED Key that is currently connected to the PED.
   If the source PED Key had an optional PED PIN assigned, then that PED PIN is automatically applied to the duplicate during this process.
8. Remove the newly imprinted PED Key and press ENTER. The PED goes back to "PED Key mode" awaiting further commands. If you wish to duplicate another PED Key, repeat the above steps. Otherwise, press "<" to go back to "Admin mode", and press "<" again to reach the main menu, and finally press "1" to resume "SCP mode", which is the normal operating mode of the PED, awaiting commands from the connected HSM.
9. Identify the new PED Key with a tag or other marker, and record a PED PIN (if any) in secure fashion, according to your security policies.

EXCEPTION: Remote PED

The Remote PED 2 functions as described above, when it is in Local or Admin mode. However, when it is placed in Remote mode, it is capable of setting up a secure connection, via a specially-configured computer workstation, to a remotely located HSM. See "Remote PED" on page 113.

How to Use a Luna PED

Luna PED, when used locally, derives its power from its connection to a Luna HSM.

To use the Luna PED:

1. Connect the Luna PED to the PED connector on the Luna HSM, using the supplied cable.
2. Luna PED performs its self-test and briefly displays its firmware version. When the display shows "SCP mode" and “Awaiting command...” Luna PED is ready to use with your Luna HSM.
3. When an activity on the server requires Luna PED operation, the Luna PED display changes, to prompt you to insert a PED Key, or to perform some other action.
4. If a PED Key is requested, remove any Key that is currently inserted (if any), and insert the requested PED Key into the USB connector slot on the right-hand top side of the Luna PED (immediately to the right of the cable connection).
5. When the Key is fully inserted, the LED in the key housing comes on.
6. Press [ENTER] on the keypad, and watch for further prompts on the display.

The Luna PED display returns to “Awaiting command..” when the current sequence of PED operations is finished. “Awaiting command..” on the Luna PED means that control has been transferred back to the HSM.

Luna PED Keypad Functions

Key	Function
[CLR] or [Clear]	- Clear the current entry, such as when inputting a PED PIN - wipes the entire entry. - *Reset the PED - the key is held down for five seconds. Useful if a PED operation has timed out.
[<]	- Backspace; clear the most recent digit that you have typed on the PED, such as when inputting a PED PIN. - "Back"; navigate to a higher-level menu in the PED.
[>]	- Shows most recent PED actions (since being in 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

Note: Pressing (and holding) [CLR] causes reset only if the PED is engaged in an operation or is actively prompting you for action.
Pressing [CLR] has no effect in the main menu, in the Admin Mode menu, or when "Awaiting command..."

Luna PED Interaction

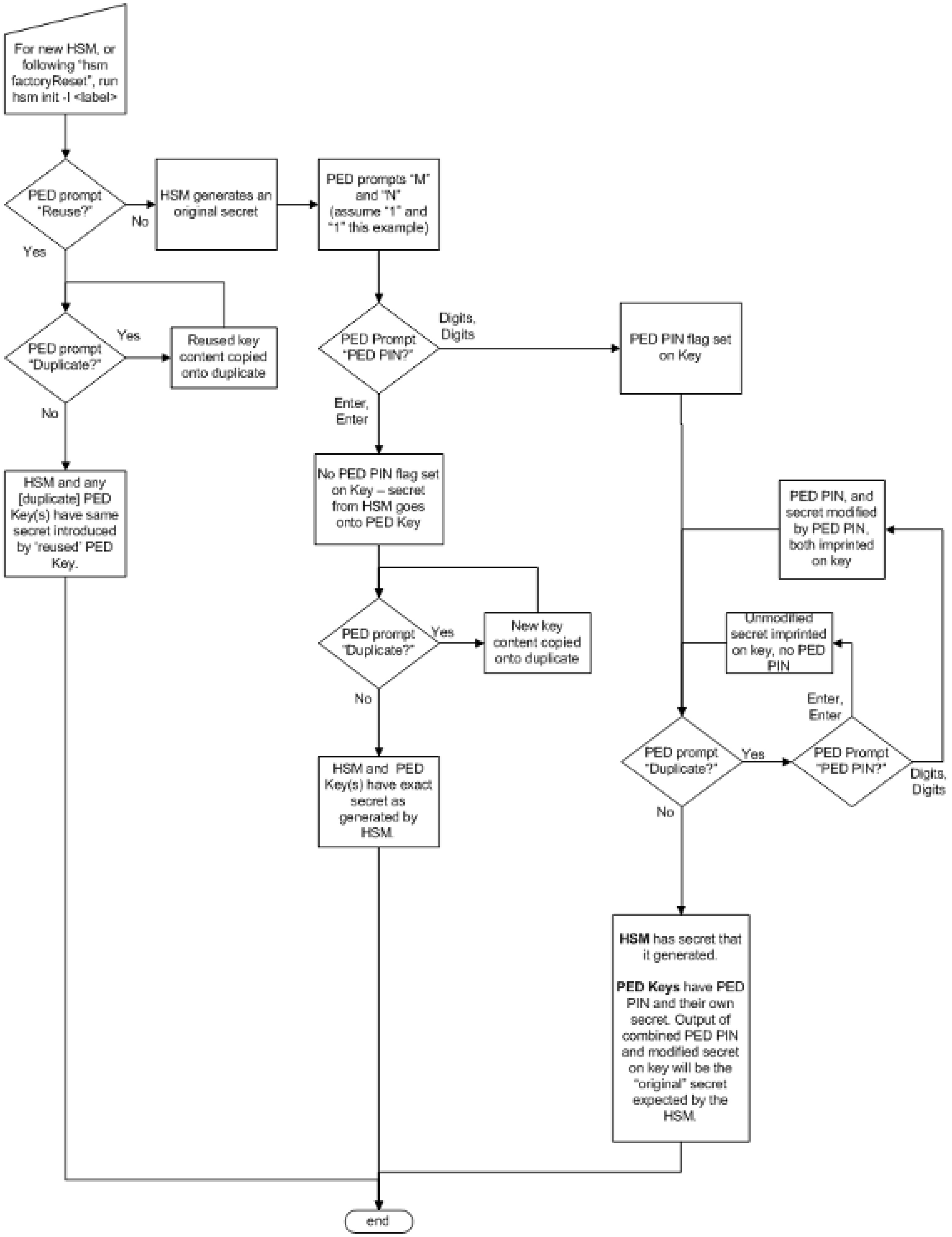
Go to "Interaction between HSM and PED" on page 72"Interaction between HSM and PED" on page 72 to read about using the Luna PED with your HSM.

Interaction between HSM and PED

(This page is background information that might help make some operations more obvious.)

After the first-ever Luna HSM, all succeeding generations have included both password-authenticated and PED-authenticated variants. This page describes how the current-generation PED-authenticated HSMs (firmware 6.x) interact with Luna PED and PED Keys, particularly during initialization - a time when important decisions are made. Other pages describe the PED and PED Keys. This is more about flow.

The diagram shows how the components are affected as you make choices during an initialization [this sequence depicts events and choices if you initialize a new, factory-fresh HSM, or one on which you have recently run "hsm factoryReset"; as well the process would be very similar for creation of a partition]. This flow depicts the SO / HSM Admin secret, but the interactions for other secrets follow the same pattern.


For new HSM, or following "hsm factoryReset", run hsm init -l <label>
PED prompt "Reuse?"
No
Yes
HSM generates an original secret
PED prompts "M" and "N" (assume "1" and "1" this example)
PED prompt "Duplicate?"
Yes
Reused key content copied onto duplicate
No
HSM and any [duplicate] PED Key(s) have same secret introduced by 'reused' PED Key.
PED Prompt "PED PIN?"
Digits, Digits
PED PIN flag set on Key
Enter, Enter
No PED PIN flag set on Key – secret from HSM goes onto PED Key
PED prompt "Duplicate?"
Yes
New key content copied onto duplicate
No
HSM and PED Key(s) have exact secret as generated by HSM.
PED PIN, and secret modified by PED PIN, both imprinted on key
Unmodified secret imprinted on key, no PED PIN
Enter, Enter
PED prompt "Duplicate?"
Yes
PED Prompt "PED PIN?"
Digits, Digits
No
HSM has secret that it generated.
PED Keys have PED PIN and their own secret. Output of combined PED PIN and modified secret on key will be the "original" secret expected by the HSM.
end

When you issue the "hsm init" command at the command-line, the HSM generates a secret, then turns over control to the Luna PED.

Reuse? (a.k.a. Group PED Key)

The first question from the PED is whether you wish to "Reuse" an existing SO / HSM Admin authentication secret (the same logic applies to the other PED Keys, so we use just the blue key in this example). This means that you have a blue PED Key from another HSM, or you have a blue PED Key from a previous initialization of this HSM. The PED is asking if you wish to import the secret from that key onto the HSM. The options at this point are:

a) you have only fresh blank PED Keys that have not been used previously with any HSM (No - do not reuse)

b) you have a previously used PED Key, but the secret it contains is not one you wish to preserve or re-use (No - do not reuse)

c) you have a previously used PED Key, with a secret from this HSM, and you don't mind reusing it (Yes - reuse)

d) you have a previously used PED Key, from another HSM, and you wish to reuse it so that the blue key can unlock both the current HSM and the other HSM. (Yes - reuse)

These options also apply to any other key color when they are being imprinted. If you elect to reuse the content of an existing key, then the secret that the current HSM generated is discarded, and the secret from the reused PED Key overwrites onto the HSM. This ensures that the PED Key and the HSM have the same authentication secret, and the key can unlock the HSM. If the secret on the key was from another HSM that is still operational, then the PED Key has become a "group PED Key" that unlocks the equivalent aspects of both HSMs. In this manner, you can include as many HSMs as you wish in a group. [Note that this "group" of HSMs is related only by the convenience of being able to use one PED Key to unlock any of them. This "group" concept is not the same as (say) the HA Group concept for high availability.

The HSM slots that form an HA group interact with their client(s) via a virtual HSM slot, such that any of the real HSM slots behind the HA group is interchangeable and can be swapped in and out as needed. But members of an HA group do not need to be members of a PED Key group. In an HA group, any or all of the members could have the same or different authentication secrets, without affecting the HA function. Only the cloning domain must be identical across all HA group members.]

If you choose to **not** reuse the content of an existing key, then the secret that the current HSM generated is copied onto the key that is currently inserted into the PED (after the PED verifies multiple times that this is what you wish to do). This ensures that the PED Key and the HSM have the same authentication secret, and the key can unlock the HSM. If the PED Key previously had a secret for another HSM, it no longer does. The PED Key can now unlock the current HSM but is useless with the previous HSM.

Note also that your organization's security policies govern whether you can allow multiple HSMs or HSM partitions to be unlockable by the same PED Key.

MofN?

The second question from the PED would ask for M and N values, so that you could set up MofN split-secret, multi-person access control. However, that option would greatly complicate this explanation, so we will assume that you choose M=1 and N=1, which means "no MofN invoked".

PED PIN?

The PED provides the opportunity to add an additional layer of authentication security to the handling of the current secret. A PED PIN is a numeric secret typed on the PED keypad. If you just press enter, no PED PIN is created, and therefore no PED-PIN flag is set on the current PED Key. If you do type in some digits on the PED's keypad, then that sequence becomes a PED PIN, a numeric password that must be typed whenever you wish to use that key in future. Whatever your response, the PED asks you to confirm by typing it in again, before proceeding to the next question.

Duplicate? (make backups)

The next question from the PED is whether you wish to duplicate the current PED keyset. [The word "keyset" is used because you could have chosen to invoke MofN, splitting the (in this case) HSM secret across several blue keys, rather than just the one in this example. That is, a "keyset" can consist of one key, containing a complete secret, or multiple keys, each containing a portion of that secret.]

In general, it is a good idea to have several PED Keys with the HSM secret duplicated, so that you can have on-site and off-site backups, and to meet your other operational considerations.

The first opportunity to make copies is at initialization time, as the PED always asks this question during the process. Your answer to the "duplicate" question determines the end of the process for the current PED Key secret.

Again, your security policies dictate how many backup copies - or other operational copies - of a PED Key should be made, and how they should be handled and maintained.

How it was - versus - how it is today

Customers who are familiar with our legacy HSM products, and who are now preparing to use Luna HSM 5.x (a firmware 6.x HSM) would observe that much of the concept and action is similar to the previous generation, but with a few important differences, described below. This would be especially important for customers who are migrating keys and HSM contents from older HSMs to the current generation.

Differences in function are driven to a considerable extent by the updating of the (optional) M of N, split-secret, multi-person access control model.

Legacy

In HSM firmware 4, the MofN concept was of a separate, self-contained single secret (on green keys, and no PED PIN), so all the other PED Key colors were just one secret each, which was a simple model that allowed certain possibilities and precluded others.

In that older model, if a PED PIN was created, it existed only in your head ("something you know), and was a transformation that you applied to the secret on the key ("something you have"), to make it into the secret on the HSM. In that model, it was *not* possible to have more than one PED PIN for (say) the SO secret on HSM1. However, it *was* possible to use that same key for another HSM (2) with a different PED PIN, because the secrets on the two HSMs didn’t have to match.

All that was needed was that whatever was on the blue key could be reliably transformed into what HSM1 wanted, and could also be transformed into whatever HSM2 wanted, by typing something on the keypad.

You could minimize the number of blue keys, while still ensuring that HSM1 and HSM2 had effectively different secrets – as long as you trusted that HSM1's SO and HSM2's SO were not going to talk to each other. But any duplicate blue keys were, indeed, exact duplicates. It was the HSMs in a group that had different secrets, not the keys. The same idea applied to the black keys.

(Key #1 + PED PIN #1) or (Key #2 + PED PIN #1) = Success on HSM1,

(Key #1 + PED PIN #2) or (Key #2 + PED PIN #2) = Success on HSM2

(Key #1 + PED PIN #1) or (Key #2 + PED PIN #1) = Failure on HSM2

(Key #1 + PED PIN #2) or (Key #2 + PED PIN #2) = Failure on HSM1

Modern

In HSM firmware 6 (Luna PCI-E 5, Luna SA 5, Luna G5), the new PED-mediated MofN-per-key-color model required some re-engineering. Additional infrastructure was needed, which makes this model incompatible with the previous method.

Functionally, in the current model, it doesn’t matter whether you choose M and N to be one (feels like no M of N) or you choose M and N to be greater than one (invoking secret-splitting) – the infrastructure is there, regardless.

One result is that the HSM takes on additional responsibility for validating splits (even if there's only the one…), and the PED Key data now has a direct relationship to the PED PIN (which is part of the validation done when the PED Key is entered). Therefore, a "duplicate" is now a slightly fuzzier concept. Each duplicate PED Key can be given a different PED PIN (or none), and can still unlock the same HSM1. BUT, if you now make a group of HSMs by initializing a second HSM (HSM2) with the same basic secret (by imprinting the new HSM from one of the duplicate PED Keys), you must use the correct PED PIN for the Key used – any other choice will fail validation. The result is that the second HSM uses the same secret as the first - which is different from the firmware 4 case.

You can optionally have split each secret (M and N greater than 1 when you initialized HSM1), which just makes the combinations more interesting to track without a good set of notes, but that doesn't change the concept… merely adds a layer.

In the following table, we illustrate your interactions with the PED as you initialize an HSM or create a partition, with a fresh secret (not reused), and then create two duplicates of the PED Key, each with a PED PIN different from the original and from each other, yet all three will unlock that HSM or that partition - to simplify this exercise, we ignore MofN. Assume that all keys are fresh blanks.

HSM1 PED prompt	**Original key, No PED PIN (your action)**	**First duplicate key PED PIN "1234" (your action)**	**Second duplicate key PED PIN "4321" (your action)**
"Do you wish to reuse an existing keyset" (creating new PED Keys during initialization)	Press [No]	n/a	n/a
Insert...	(insert a new key)	-	-
Enter new PED PIN / Confirm new PED PIN	Press [Enter]	n/a	n/a
"Are you duplicating this keyset?"	Press [Yes]	-	-
Insert...	-	(insert a new key)	-
Enter new PED PIN / Confirm new PED PIN	-	Type "1234" and press [Enter]	
"Are you duplicating this keyset?"	-	Press [Yes]	-
Insert...	-	-	(insert a new key)
Enter new PED PIN / Confirm new PED PIN	-	-	Type "4321" and press [Enter]
"Are you duplicating this keyset?"			Press [No]
All three PED Keys have different PED PINs, but any one of them can unlock this HSM. The combination of any of those PED Keys, with its own PED PIN will produce the same secret for the HSM.			

To round out the parallel concept that finished the firmware 4 discussion above, any duplicate blue keys are not necessarily exact duplicates, they just all contain a way (PED PIN secret) to get back to the same output secret. But in this model (firmware 6), if you want to use the same blue keys for several HSMs, all the HSMs must have exactly the same blue (SO) secret, because a duplicate of any blue key CAN have whatever PED PIN you choose (or none) but must still be able to generate the correct secret.

(Key #1 + PED PIN #1) or (Key #2 + PED PIN #2) = Success on HSM1

(Key #1 + PED PIN #1) or (Key #2 + PED PIN #2) = Success on HSM2

(Key #1 + PED PIN #2) or (Key #2 + PED PIN #1) = Failure on HSM1

(Key #1 + PED PIN #2) or (Key #2 + PED PIN #1) = Failure on HSM2

Restating the "obvious"?

Some important implications of the above explanations deserve restating.

- If you choose to NOT reuse a secret from an existing PED Key, then the HSM and the new set of PED keys being created by initialization all receive secrets based on the secret that is newly generated by the HSM. This is how you ensure that no other HSM can be unlocked by the PED Key(s) that you are now associating with the current HSM. This exclusivity lasts as long as nobody initializes yet another HSM using the PED Key(s) that you just created for this current HSM.
- It is crucially important to always control your PED Keys. Know where they are, and know who is handling them.
- If you choose to reuse a pre-existing secret, then the secret that the HSM generates at the start of initialization is discarded, in favor of the imported secret[1]. This is how you make group PED Keys that can unlock more than one HSM.
- The PED PIN, if you invoke one, exists only in your head[2] not on the PED Key - it is the combination of the secret on the key, plus the PED PIN for that key, that produces the secret that the HSM sees (and requires).

An additional question that is sometimes asked, about reuse and duplicates...

- You can "reuse" an existing secret only for the same type of secret that is currently being requested by the HSM and the PED. That is, if you say [Yes] to "Would you like to reuse an existing keyset" while preparing to set the HSM's Security Officer (SO)/Administrator secret, then you must present a valid, imprinted blue PED Key. Any other color, or a blank key, is rejected as a source to reuse. A User (black key) secret cannot be "reused" as an HSM SO (blue key) secret. Nor can a Domain (red), or SRK (orange), or Audit (white), or SRK (purple key) secret. "Reuse" is the opposite of overwrite. For the "reuse" option, with any PED Key secret, the matching kind of pre-existing secret is needed. SRKs, the purple key secrets, are unique per HSM and are not reused, ever.

Duplicating PED Keys / Copying PED Keys

Luna PED has the ability to make copies of PED Keys, without the intervention of an HSM. All the PED needs is power. Insert any PED Key containing a secret that you wish to duplicate. The PED defaults to the local mode menu.
Press "<" to get to the Select Mode menu.
Press "4" for the Admin menu.
Press "1" for PED Key.
Press "1" again, for Login.
Press "7" for Duplicate. The PED reads the key that you already inserted, then prompts you:

[1] [the secret that you accept from an existing imprinted PED Key when you say [Yes] to the PED question "Would you like to reuse an existing keyset?"]

[2] [or wherever you write it down]

Duplicate PED Key...
Insert target
PED Key.
Press ENTER.

When you press ENTER, the key in the slot gets the data that was read from the first key.

You can imprint as many new PED Keys as you wish.

Note that the PED does NOT prompt you for a PED PIN.
If the PED PIN flag was not set on the source key (the first key you inserted before invoking the Duplicate function), then the new copy also has that flag unset.
If the PED PIN flag was set on the original key, then that setting is automatically recorded on the duplicate. No HSM is involved in this PED-only transaction, so entering a PED PIN would have no effect in this case. Yet the correct PED PIN will be requested when you later use one of these duplicates to access the HSM.

This DIFFERS from the sequence when you are initializing and choose to make duplicates at that time - in that case you are prompted for PED PIN and can make several "duplicate" keys that have different PED PINS and yet unlock the same HSM. This method is called a "raw" duplication and works for every type of PED Key except a purple SRK.

Comparing Duplication via PED menu - versus - "Duplication" when initializing

	Requires HSM	**Launched from command line**	**Prompt (option) to set PED PIN**	**"Copies" are identical**	**"Copies" unlock same HSM**
"Duplicating" (creating new PED Keys during initialization)	Yes	Yes	Yes	Only if no PED PIN or if same PED PIN is repeatedly entered	Yes, as long as you know the correct PED PIN for the key you have
Duplicating "raw" key content via PED menu	No (only a power connection needed) Note: does not work for purple PED Key.	No	No	Yes	Yes, PED PIN is the same for all raw duplicates

Lost PED Keys, PED PINs, or Passwords

Help! I have lost my blue/black/red/orange/purple/white PED Key or I have forgotten the password!

ANSWER-general (Passwords): Go to the secure lockup (a safe, an off-site secure deposit box, other) where you sensibly keep such important information, read and memorize the password. Return to the HSM and resume using your HSM(s).

ANSWER-general (PED Keys): Retrieve one of the copies that we (and your security advisor/consultant) always advise you to make, from your on-site secure storage, or from your off-site [disaster-recovery] secure storage, make any necessary replacement copies, using Luna PED, and resume using your HSM(s).

If you have lost a blue PED Key, someone else might have found it. Consider `lunacm:>changePw` or `lunash:>hsm changePw`, as appropriate to invalidate the current blue key secret, which might be compromised, and to safeguard your HSM with a new SO secret, going forward. HSM and partition contents are preserved.

But I don't have keys or secrets in secure on-site or off-site storage! What do I do?

ANSWER - blue PED Key or SO password : If you truly have not kept a securely stored written backup of your HSM SO Password, or for PED-authenticated HSM, your blue SO PED Key, then you are out of luck. If you **do** have access to your partition(s), then immediately make backups of all partitions that have important content. When you have done what you can to safeguard partition contents, then perform `hsm factoryReset`, followed by `hsm init` - this is a "hard initialization" that wipes your HSM (destroying all partitions on it) and creates a new HSM SO password or blue PED Key. You can then create new partitions and restore contents from backup. Any object that was in HSM SO space (rather than within a partition) is irretrievably lost.

ANSWER - black PED Key or Partition User password : If you truly have not kept a secured written backup of your partition User Password, or for PED-authenticated HSM, your black partition User PED Key, then log into your HSM as SO, and perform `partition resetPw`. The `partition changePw` action is done by a partition owner who has the current credential and wishes to change it, so that one is not available to you now. The `partition resetPw` is done by the HSM SO when the current partition secret has been lost, or is compromised (perhaps by the unplanned departure of personnel). Select option 4 when you run the command.

```
lunash:> partition resetpw -partition mypar

Which part of the partition password do you wish to change?

1. change User or Partition Owner (black) PED key data
2. generate new random password for partition owner
3. generate new random password for crypto-user
4. both options 1 and 2

0. abort command

Please select one of the above options: 4

Luna PED operation required to reset partition PED key data - use User or Partition Owner (black) PED key.

****

'partition resetPw' successful.

Command Result : (Success)
lunash:>
```

**** Follow the PED prompts:
a. press [No] when asked “Would you like to reuse an existing keyset? (y/n)”

b. provide the M and N values of your choice ([1] and [1] if you don't want MofN)

c. press [Yes] to overwrite the user key

d. provide your choice of PED key PIN when prompted (or just press [Enter] if you do not wish to impose a PED PIN)

e. press [Yes] when asked “Do you want to duplicate the keyset? (y/n)”

f. write down the new random challenge from the PED screen (for best legibility, type it)

Now that you have the new partition authentication, you can change the PED-generated text challenge to something more to your liking via the `partition changePw` command, choosing option 3.

```
lunash:> partition changePw -partition mypar1

Which part of the partition password do you wish to change?

1. change partition owner (black) PED key data
2. generate new random password for partition owner
3. specify a new password for the partition owner
4. both options 1 and 2

0. abort command

Please select one of the above options: 3
> ****************

Please enter the password for the partition:
>********

Please enter a new password for the partition:
>********



'partition -changePw' successful.

Command Result : 0 (Success)
lunash:>
```

ANSWER - red PED Key or HSM-or-Partition domain secret: If you have the red PED Key or the HSM-or-Partition domain secret for another HSM or Partition that is capable of cloning (or backup/restore) with the current HSM or Partition, then you have the domain that you need - just make a copy. Cloning or backup/restore can take place only between entities that have identical domains, so that other domain must be the same as the one you "lost".

If you truly have not kept a secured written backup of your HSM or partition cloning domain, or for PED-authenticated HSM, your domain PED Key(s), then you are out of luck. Any keys or objects that exist under that domain can still be used, but cannot be cloned or backed-up or restored. You have no fall-back, in case of accident. Begin immediately to phase in new/replacement keys/objects on another HSM, for which you DO have the relevant domain secret(s) or red PED Key(s). Ensure that you have copies of the red PED Keys, or that you have a written record of any text domain string, in secure on-site and off-site backup locations. Phase out the use of the old keys/objects, as you have no way to protect them against a damaged or lost HSM.

ANSWER - orange PED Key : You will need to generate a new Remote PED Vector on one affected HSM with `lunacm:>ped vector init` or `lunash:>hsm ped vector init` to have that HSM and an orange key (plus backups) imprinted with the new RPV. Then you must physically go to all other HSMs that had the previous (lost) RPV and do the same, except you must say "Yes" to the PED's "Do you wish to reuse an existing keyset?..." question, in order to bring the new RPV to all HSMs that are intended to use Remote PED with the new orange PED Key(s). If you forget and say "No" to the PED's "...reuse..." question, then you are starting over.

ANSWER - white Audit PED Key : You will need to initialize the audit role on any affected HSM. This creates a new Audit identity for that HSM, which orphans all records and files previously created under the old, lost audit role. The audit files that were previously created can still be viewed, but they can no longer be cryptographically verified. Only records and files that are created under the new audit role can be verified, in future. Remember, when performing Audit init on the first HSM, you can say "Yes" or "No" to Luna PED's "Do you wish to reuse an existing keyset?..." question, as appropriate, but for any additional HSMs that must share that audit role, you must answer "Yes" to "Do you wish to reuse an existing keyset?..."

ANSWER - purple PED Key : If SRK was not enabled, this is not a problem - any purple PED Keys you had for that HSM are invalid anyway. If SRK was enabled, then your options depend on whether the HSM is currently in a tamper condition or Secure Transport mode... or not. There is no way to recover from a tamper or from Secure Transport Mode if the external split of the Master Tamper Key (the SRK) is not available. If you haven't got a backup purple key, your HSM is locked the moment it experiences a tamper event, or if it was placed in Secure Transport Mode. The same applies if you do have the key, but have forgotten/lost a numeric PED PIN that you [optionally] applied when the purple key was imprinted with the Secure Recovery Vector (the external split of the MTK). Either way, you must obtain an RMA and return the HSM to SafeNet for remanufacture. All HSM contents are lost.

If the purple key is lost, BUT the HSM is still in working mode - that is, it has not experienced a tamper event, and you have not placed it in Secure Transport Mode - then you should immediately rescue any important HSM or partition contents by backing them up, and restoring onto another HSM (that does NOT have SRK enabled, or for which SRK is enabled, but you DO still have the purple key). Once that is accomplished, decommission the original HSM, obtain an RMA, and ship it back to SafeNet for re-manufacture. It is not safe to continue using an HSM that has SRK enabled, but for which you have lost the purple PED Key. Any tamper event would render contents irretrievable. Avoid putting yourself in such a situation.

I have my PED Key, but I forgot my PED PIN! What can I do?

Forgetting a PED PIN is the same as not having the correct PED Key. See above, for your options in each situation. A PED PIN is an [OPTION] that you decide, at the time a role is created. If your security regime/protocol demands that your HSM access must enforce multi-factor authentication, then a PED PIN is a useful/necessary option for you. If your security protocol does NOT demand such measures, then you should seriously consider whether it is justified.

Once a PED PIN is imposed, it is a required component of role authentication, until/unless you arrange otherwise. You can remove the requirement for a PED PIN on a given HSM role only if you are currently able to authenticate (log in) to that role. For black PED Keys, you can have the SO reset your authentication. For other roles... not.

Thus, for blue or purple PED Keys, forgetting a PED PIN, like losing the PED Key (with no backups) is fatal.

For red PED Keys, forgetting the PED PIN is eventually fatal, but you can work in the meantime while you phase out your orphaned keys and objects.

Forgetting PED PINs for other roles, like losing their PED Keys is just more-or-less inconvenient, but normally not fatal.

I have my PED Keys and my PED PINS, but I can't remember which one goes with which HSM (or partition)!

See your options, above. The most serious one is the blue PED Key or the PED PIN for the SO role. You have only three tries to get it right. On the third wrong attempt, the HSM contents are lost. Wrong attempts are counted if you present the wrong blue PED Key, or if you type the wrong PED PIN with the right PED Key.

For black User PED Keys, and their PED PINS (if applicable) you have ten tries to get the right key or the right combination, unless the SO has changed from the default number of retries. If you are getting close to that maximum number of bad attempts, stop, and ask the SO to reset your partition PW.

For other PED Keys, there is no restriction on re-tries. Good luck. Try to be better organized in future.

Commands that Require Luna PED Interaction

The following is a list of HSM commands that require the use of Luna PED and PED Keys for PED-authenticated Luna HSMs. These indications apply whether you are using local PED or Remote PED [*].

Commands (hsm)	Notes
hsm login	PED required SO (blue) key
hsm changeHSMPolicy	Must be logged in as SO to complete. PED required - SO (blue) key
hsm changeSOPolicy	Must be logged in as SO to complete. PED required - SO (blue) key
hsm changePw	PED required SO (blue) key
hsm contents	Must be logged in as SO to complete. PED required - SO (blue) key
hsm clear	Must be logged in as SO to complete. PED required - SO (blue) key
hsm updateFW	Any partition activated before firmware update will need to be reactivated after the update - par activate command requires use of PED
hsm rollbackFW	Must be logged in as SO to complete. PED required - SO (blue) key
hsm updateCap	Must be logged in as SO to complete. PED required - SO (blue) key
hsm restoreSIM2	Must be logged in as SO to complete. PED required - SO (blue) key
hsm restoreUser	Must be logged in as SO to complete. PED required - SO (blue) key
hsm clone	Must be logged in as SO to complete. PED required - SO (blue) key
hsm restore	Must be logged in as SO to complete. PED required - SO (blue) key
hsm factoryReset	Technically a PED is not required to execute this command (a direct serial connection to the device is required), but a PED will be required when re-initializing and configuring the HSM
hsm smkClone	Must be logged in as SO to complete. PED required - SO (blue) key
hsm setLegacyDomain	Legacy Domain

	PED key required, therefore the PED is required

Commands (partition)	**Notes**
par login	User password required PED required User (black) key
par activate	User password required PED required User (black) key
par create	Must be logged in as SO to complete. PED required - SO (blue) key PED required User (black) key PED required Domain (red) key
par createUser	Must be logged in as SO to complete. PED required - SO (blue) key
par createChallenge	Must be logged in as SO to complete. PED required - SO (blue) key PED also displays the generated challenge string
par changePolicy	Must be logged in as SO to complete. PED required - SO (blue) key
par changePw -p	Must be logged into partition User password required PED required User (black) key Old and new password required
par resetPw	Must be logged in as SO to complete. PED required - SO (blue) key PED required User (black) key
par contents	User password required PED required User (black) key
par clear	User password required PED required User (black) key
par backup	User password required PED required User (black) key
par clone	User password required PED required User (black) key
par setLegacyDomain	User password required PED required User (black) key
par restoreSIM2	Must be logged in as SO to complete. PED required - SO (blue) key

par restoreSIM3	Must be logged in as SO to complete. PED required - SO (blue) key

Commands (srk)	Notes
srk enable	Must be logged in as SO to complete. PED required - SO (blue) key
srk disable	Must be logged in as SO to complete. PED required - SO (blue) key
srk recover	Must be logged in as SO to complete. PED required - SO (blue) key
srk generate	Must be logged in as SO to complete. PED required - SO (blue) key
srk transport	Must be logged in as SO to complete. PED required - SO (blue) key

[* The only instance where local and Remote PED operations are not equivalent is when you initially set up for Remote PED operation by imprinting an RPV (Remote PED Vector) using an RPK (orange Remote PED Key). The imprinting must be performed locally.

Once the HSM has an RPV, you can perform all further PED-mediated authentication remotely, if desired.]

In most cases, use of the PED is a rare event. You use it when setting up the HSM and partitions, when activating partitions, and when making certain changes that might be necessitated by changes or expansions in your application or security environment (example: change of personnel). Most customers find that, once provisioned for your environment and application(s), the HSM simply functions day after day with no further intervention required.

CHAPTER 9
PED Key Management

This chapter describes how to manage your PED keys. It contains the following sections:

PED Key Management Overview

This section applies to Luna HSMs with PED (Trusted Path) Authentication, only.

As indicated elsewhere, the capability to imprint “group-User” PED Keys and “duplicate-User” PED Keys, permits considerable flexibility in the use, archiving and general management of PED Keys.

The following pages address the ongoing management of PED Keys (which would normally include at least one "working" or "production" set, and at least one backup set, possibly stored off-site).

When you initialize an HSM or create a Partition, Luna PED prompts you for various PED Keys and actions. Some are mandatory, some are advisable, and some are optional, depending upon your situation and requirements. Here is a quick summary:

Imprint a Blue PED Key

When an HSM is initialized, it sets up a blue Security Officer (SO) or HSM Admin authentication PED Key (two names for the same function, depending upon the industry you are in). This is the key that you will need in future, to access that HSM. This can be done in one of two ways:

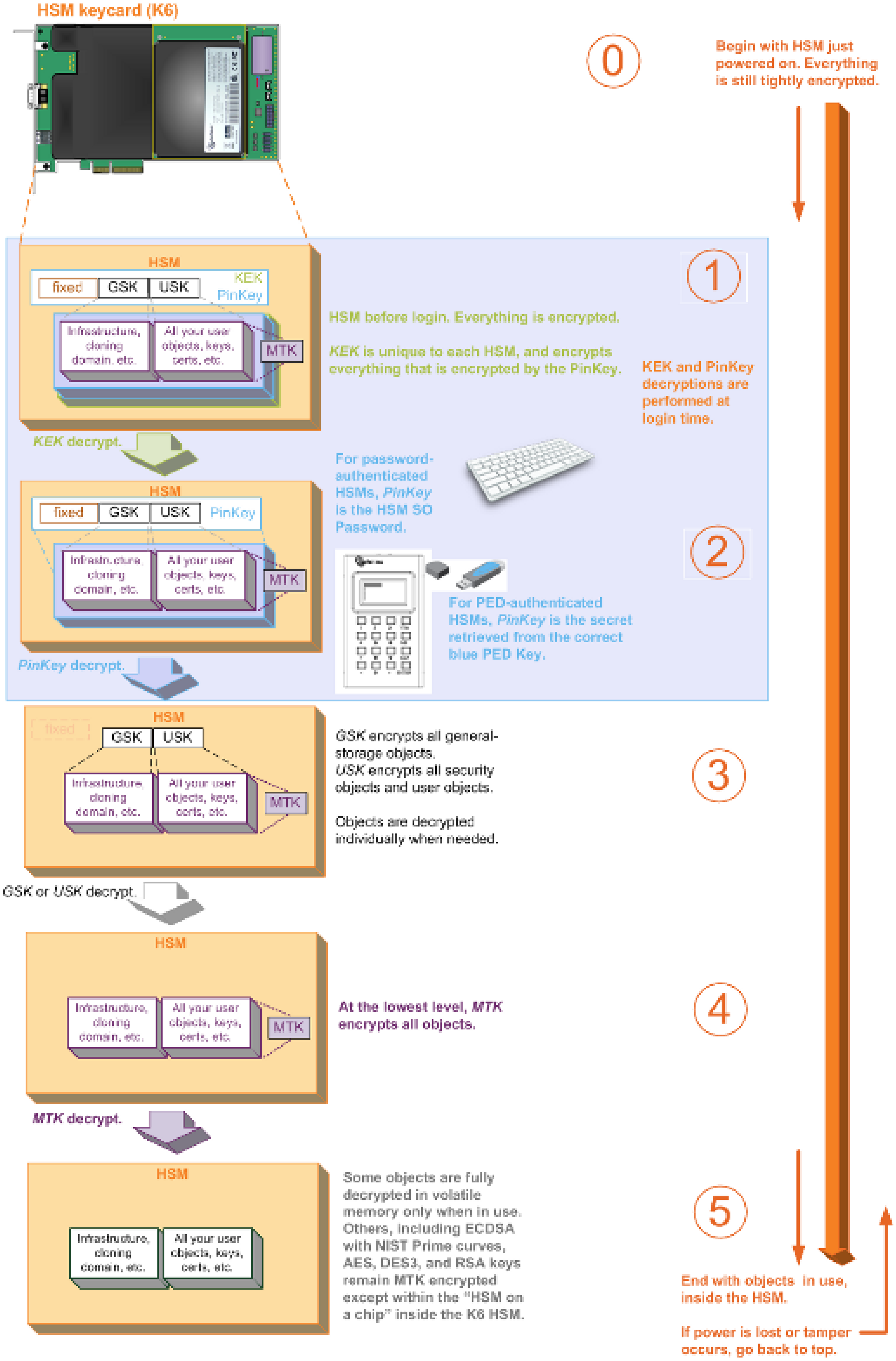
- the HSM can generate new, unique, random authentication data and imprint it onto a blue PED Key -- the resulting blue PED Key will now unlock that HSM, but no other
(you do this when you answer "NO" to the "reuse an existing keyset (roughly equivalent to the "Group PED Key" question on the old PED 1.x)" question from the Luna PED)

OR

- the HSM can read the authentication from a blue PED Key that was already imprinted by another HSM, and accept that data as its own -- the blue PED Key can now unlock two (or more) different Luna HSMs
(you do this when you answer "YES" to the "Reuse an existing keyset" question from Luna PED)

-

HSM Layered Encryption - the General Case



Begin with HSM just powered on. Everything is still tightly encrypted.

HSM before login. Everything is encrypted.

KEK is unique to each HSM, and encrypts everything that is encrypted by the PinKey.

KEK and PinKey decryptions are performed at login time.

For password-authenticated HSMs, *PinKey* is the HSM SO Password.

For PED-authenticated HSMs, *PinKey* is the secret retrieved from the correct blue PED Key.

GSK encrypts all general-storage objects.
USK encrypts all security objects and user objects.

Objects are decrypted individually when needed.

At the lowest level, *MTK* encrypts all objects.

Some objects are fully decrypted in volatile memory only when in use. Others, including ECDSA with NIST Prime curves, AES, DES3, and RSA keys remain MTK encrypted except within the "HSM on a chip" inside the K6 HSM.

End with objects in use, inside the HSM.

If power is lost or tamper occurs, go back to top.

During initialization of an HSM, the HSM determines which blue PED Key will "unlock" the HSM in future. The HSM can create new, random authentication data and imprint that data onto a blue PED Key, **or** the HSM can scan an existing (previously imprinted) blue PED Key from another HSM and set the data from that older blue key as the new HSMs own "unlocking" data.

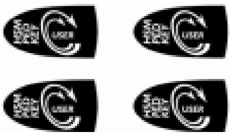
- For your very first HSM, you **must** initialize a blue PED Key for the HSM Admin.
- If this HSM is not the first; if you are creating a group of HSMs that are related in some way, then you CAN initialize a new blue PED Key for it, or you can re-use the authentication data on another blue PED Key (by deciding it will be a group PED Key - see "Shared or Group PED Keys" on page 90). This is your option. The HSM requires an imprinted blue PED Key when you access it, but you decide (at HSM initialization) whether that blue PED Key should be unique to this particular HSM, or shared among two or more HSMs.
- Whenever you perform an initialization, the Luna PED also gives you the option to make duplicates of your important PED Keys. If you already have enough (at least one primary and at least one backup), then you can just answer "NO" to the "Copy this key" prompt. If you need more of the current type of PED Key (in this case, the blue HSM Admin PED Key), then say "YES" and continue supplying additional blank keys until you have enough duplicates.

If you are new to using PED keys and your security policy allows it, you should make a duplicate copy of the blue Security Officer and red cloning domain PED Keys as backups. And please review "General Advice on PED Key Handling" on page 103 at this time.

PED Keys and Operational Roles

Below are some suggested holders of PED Keys by role.

Lifecycle	PED Key	Operational Role	Function	Custodian
PED keys enforce division of operational roles and prevent unilateral action by key holders				
Admin	 or (*) 	Security Officer	Token/HSM Administration Set token security policy Select token initialization parameters Create Users	CSO CIO
	 or (*) 	Domain Cloning Token Backup	Set Cloning Policy Create/Transfer Cloning Domains Token Backup	Domain Administrator WAN Administrator

Lifecycle	PED Key	Operational Role	Function	Custodian
	 or (*) 	Secure Recovery	Recover from Secure Transport Mode or tamper event	CSO
	 or (*) 	Remote PED	Establish a Remote PED connection	System Administrator
Daily Operation	 or (*) 	HSM Partition User or Partition Owner	Key Generation Signing Decryption	System Administrator
Ongoing Auditing	 or (*) 	Audit User	HSM Audit logging HSM Audit Archiving	Auditor

[*In each case, you have the option, instead of a single PED Key of a particular color and function for a particular role, to *split* the relevant authentication secret across several PED Keys of that color and function (we show 4, in the table, but you could have any number of splits up to 16 for each secret). This is the M of N split-knowledge shared-secret option that you set when a PED Key is created. The M of N option is set, or not, as a result of your choices when responding to PED prompts for "M value" and "N value".

Set "M" and "N" to 1, and no split occurs. The secret - blue SO key, red Domain key, black User or Partition Owner key, purple Secure Recovery key, white Audit Key, or orange Remote PED key - is not split, and is imprinted on just a single key of that color. The result is that a single person can hold/control the entire authentication authority for that role.

Set "M" and "N" to values higher than 1, and the secret gets split across "N" different keys of the relevant color. A minimum of "M" of those keys must be brought together in order to authenticate. This allows the role to be spread over multiple persons, with no one holder able to authenticate by himself/herself.

The multiple keys of a split secret are NOT to be confused with DUPLICATE or COPY keys. During key creation (and at other times) the PED enables you to make extra copies of the complete secret for, say, the SO or the Partition User. Those are identical copies that you can

- store for offsite backup,
- distribute to personnel on other work-shifts, or
- apply to whatever use your security policies permit/direct for such copies.

If you *do not* set M of N (example N = 1), and you ask for copies, then the PED makes 1 copy of the 1 key that holds the complete secret. You can make as many copies as you wish, but each one alone is capable of unlocking that function/role on the HSM. No other person is needed in order to unlock and use that function/role of the HSM. No other person knows about that usage, except after the fact if you have set up Audit Logging.

If you *do* set M of N (example, 3 of 5), and ask for copies, then the PED makes 1 copy of each of the 5 original splits. Each split within a set is unique in that set. Authenticating with M of N requires that you present M different splits to reconstitute the authentication secret. Avoid mixing members of M of N sets.To illustrate why this is bad, if your SO authentication was split over N=5 keys and needed M=3 different splits to authenticate, and you presented split-1 and split-2 from the original set, along with split-1 from a copy set, authentication would fail. From the perspective of the PED, you have tried to present split-1 twice, rather than providing three different splits from the secret. The PED would rightly refuse to authenticate.

Because M of N is controlled on the PED, you can choose to have some secrets split and others not split. You could have just one blue key needed for SO administrative actions, but (say) 3 black keys needed for each partition activation.
The point is to use M of N where it is important that a single person not be able to perform that role without supervision/participation by other co-key-holders. Each organization can have its own rules in that respect.

The M of N trade-off is greater security for greater management load. Split secrets increase security by requiring agreement and participation among multiple persons and thereby preventing unilateral action by a single person. But split secrets mean the existence and tracking and management of more physical keys and holders of those keys.]

Shared or Group PED Keys

With the common administrative group option (answer "YES" to the Luna PED question **Do you wish to reuse an existing keyset?** during HSM initialization or Partition creation) (one PED Key accesses multiple HSMs) – as opposed to the default unique secret (where each HSM has its own unique PED Key) – you can use numerous HSMs and not need to manage numerous keys.

For example, at an installation employing five Luna HSMs:

- the unique key option would create five different, mutually exclusive blue SO PED Keys, one to access each of the individual HSMs (a gain in exclusivity of HSM ownership, at the cost of additional PED Keys to manage and control)

  compared to

- the common administrative group PED Key option where you might have a single SO PED Key that could access any of the five HSMs (a savings at the administrative level, at the cost of HSM ownership exclusivity (if one key is compromised, it compromises all five HSMs)).

How does it work?

During the process of initializing an HSM, or creating an HSM Partition (on Luna HSM with PED [Trusted Path] Authentication), Luna PED attempts to imprint a blue or a black or a red PED Key [Similarly, the orange PED Key can be shared among several HSMs, although it is created in its own process, and not as part of HSM initialization or partition creation. The white Audit PED Key is also created and maintained in its own process, and not as part of HSM or partition initialization. Both the orange and white keys, like the others, can be made common among multiple HSMs if desired.

The purple PED Key is unique in that it can correspond to **one** HSM only.], and asks:

Do you wish to reuse an existing keyset?

Press “YES” on the Luna PED keypad if you are inserting a key that can access previous HSMs (meaning that another HSM was initialized with this PED Key). Choosing “YES” *preserves* the old access code on the PED Key and applies it also to the current HSM or token. Thereafter, the PED Key can access both (or multiple) HSMs or tokens that share the same access secret. The randomly-generated PIN on the PED-key is not overwritten.

In other words, saying "YES" to the PED prompt "Do you wish to reuse an existing keyset", is the method to share a common authentication secret among multiple HSMs.

Alternatively, if you wish to have different PED Keys associated with each HSM in your possession, answer 'NO'. A 'NO', is a choice to overwrite the PIN (if one is already present) and store a new, randomly-generated PIN on this PED Key – any existing authentication code on this PED Key is to be overwritten with a new code, good with only the current HSM or token. The same applies to black HSM Partition User PED Keys.

The red PED Keys **must** have the same domain secret for each HSM that will synchronize (backup and restore, or HA) with another. An HSM backup partition or token content can be restored only onto an HSM that was initialized with the same red key secret. You must always choose to "...reuse an existing keyset" when initializing any HSM after the first one in a cloning group, or any partition after the first one in a cloning group.

The orange RPK PED Key, for RPV (Remote PED Vector), carries a secret that matches the RPV on an HSM to which you will be remotely authenticating with Luna PED 2 remote version. If you wish more than one HSM to have the same RPK, then you would choose to "...reuse an existing keyset" when setting RPK with "hsm ped vector init".

The white Audit PED Key carries the secret that authenticates the holder of the Audit role for the current HSM, and for any other HSMs where you have chosen to "Reuse" the PED Key when initializing the Audit role.

Reusing a PED Key forces all PED PINS to be the same

The Exception

The purple SRK PED Key differs from the others, in that it cannot be used with more than one HSM in common. You can reuse a purple PED Key with a different HSM by overwriting the key, but you cannot reuse the secret on that key with any HSM other than the one that originated the secret. The SRV (secure recovery vector) is not transferable. Each SRV is unique. An HSM can export a split of its SRV onto a purple PED Key (SRK) for use with only that HSM. If you imprint a valid purple PED Key with any other HSM, the key takes on a new SRV split that is valid with the new HSM, and is no longer useful with the original HSM.

Domain PED Keys

A domain PED key is an iKey 1000 (marked with)

and imprinted with a domain secret.

A domain PED Key (the red one) carries the key-cloning vector (the domain identifier) that allows cloning to take place among HSMs and tokens. Cloning is a secure method of copying HSM (or Partition) or token objects, such that they can be replicated between HSMs and tokens, but:

- strongly encrypted (never in the clear), and
- only between HSMs and tokens that share a cloning domain.

Cloning is the method by which secure HSM and Partition backup is possible to a Luna Backup HSM, and by which restoring is possible from a Backup HSM or token to a Luna HSM or Partition. It is also used when HSM log records and files are verified by an HSM other than the one that originally created those records.

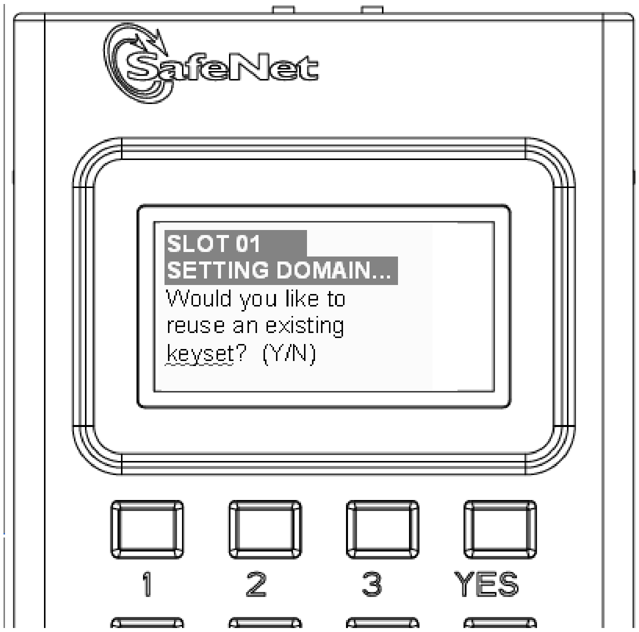
At initialization time, the key-cloning vector is created on the HSM and imprinted onto a red PED Key, or if a desired cloning domain already exists, then the existing key-cloning vector from a red PED Key is read from that PED Key and imprinted on the HSM (or Backup token) as the HSM (or token) is initialized. HSMs and tokens that share a key-cloning vector are said to be members of a cloning domain.

An HSM or token can be a member of only one domain. To make an HSM or token become a member of a second or different domain, you must initialize the HSM or token and imprint the new key-cloning vector -- the first one is destroyed and the HSM or token is now a member of only the second domain. This action also destroys any previous content on the HSM being initialized.

To cause a Luna HSM or Partition to be a duplicate or mirror image of another, the procedure is to backup the first HSM or Partition, and then restore from the Backup token onto the new HSM (or Partition).

The "New Domain" Question

When you initialize an HSM, and are prompted for a red PED Key, Luna PED first asks:



If you answer [No]:

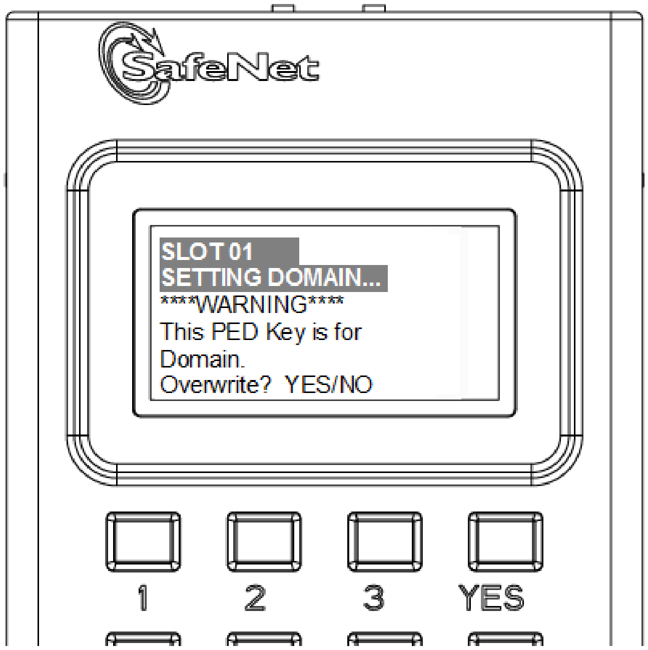
- You are telling Luna PED that it should retrieve a new domain (Key Cloning Vector) from the HSM and prepare to overwrite that new domain secret onto a blank key that you are about to insert, or overwrite the existing random domain vector on a red PED Key that you are about to insert.
- This was your last chance (short of aborting the procedure) to make the current HSM part of an existing cloning group. Further prompts in this sequence will give you the opportunity to remove keys that you have mistakenly offered (that have useful authentication secrets on them) and substitute another, but you get no more opportunity to change the "No" to a "Yes".
- If that red PED Key was already in use on an operational HSM (and Backup HSM), then that HSM (as well as the backup) carries the old domain and the newly overwritten red PED Key can no longer be used with it — therefore, unless you have a duplicate red PED Key with the old cloning domain (key-cloning vector), then that previous HSM cannot be backed up, and its Backup cannot be restored

If you answer [Yes]:

- Luna PED prepares to preserves the domain (key-cloning vector) value that it now expects to find on the red PED Key, and store it onto the HSM -- this causes the current HSM to share the domain with the previous HSM and/or Backup HSM
- With two or more HSMs (and at least one Backup HSM) sharing the same cloning domain, it is possible to clone the contents from one to another by means of backup and restore operations

Assuming that you responded [No], the PED asks additional preparatory questions, then asks you to insert a PED Key (which you should already have labeled with a red sticker). The PED scans the red PED Key for an existing key-cloning vector. If none is found, Luna PED imprints a new one, taken from the HSM, and that same new key-cloning vector is saved onto the HSM.

However, if an existing key-cloning vector (or other secret) *is* found, Luna PED needs to know whether to retain it. Luna PED asks:



SLOT 01
SETTING DOMAIN...
****WARNING****
This PED Key is for
Domain.
Overwrite? YES/NO

If you answer Yes:

- Luna PED overwrites the existing random domain vector (or other secret) on the inserted red PED Key
- If that red PED Key was already in use on an operational HSM (and Backup token), then that HSM (as well as the token) carries the old domain and the newly overwritten red PED Key can no longer be used with it — therefore, unless you have a duplicate red PED Key with the old cloning domain (key-cloning vector), then that previous HSM cannot be backed up, and its Backup token cannot be restored

If you answer No:

- Luna PED goes back a step and asks you to "Insert a Domain PED Key", which is your opportunity to correct the mistake by removing the first PED Key and inserting either a fresh (never-imprinted PED Key, or inserting a PED Key that contains an outmoded secret (Domain, SO, User, RPV, SRV).
- Each time you insert a PED Key, during an operation that could write to the key, Luna PED tells you if it is blank or if it contains a pre-existing secret, and asks if you wish to overwrite. This continues until you insert a key and allow the PED to overwrite whatever is-or-isn't on that key, or until the operation times out.
- If two or more HSMs (and at least one Backup HSM) share the same cloning domain, it is possible to clone the contents from one to another by means of backup and restore operations

To What Does a Domain Apply?

Each HSM has a domain that covers any object that can exist in the SO space - this is created at HSM initialization time. Usually objects in the SO area of the HSM are specialized keys used to facilitate HSM operations (example, masking key).

Each partition in an HSM has a domain of its own - this is created when the partition is created/initialized. Partitions contain customer-owned keys used in client operations, as well as data objects.

Objects on a partition can be cloned to another partition (whether on the same HSM or on another HSM) only if both partitions share the same domain.

In the current Luna HSM 5.x sense, one domain is like another [there is nothing special about one firmware 6 domain versus another firmware 6 domain] and could be applied to any partition or HSM SO space. Only your security and management policies dictate how you share domains. You can segregate HSMs and partitions into clonable groups. Cloning can occur among any/all members of a group that share a domain. Cloning cannot occur between members of two different domain groups.

Any HSM SO space can have only one domain, assigned at initialization time.

Any partition can have only one domain, assigned at partition creation time. It is not possible for a partition or an SO space to be a member of more than one domain. It is possible for different partitions on the same HSM to be members of mutually exclusive domains (applies to certain Luna HSM products, only).

There is no limit to the number of partitions or HSMs that can share a common domain.

What about Legacy HSMs and Partitions?

HSMs before the K6 (the HSM inside Luna SA) and G5 (the HSM for PKI with Luna SA, and the core of the Luna Backup HSM) - legacy HSMs - used an older, smaller domain secret, which is incompatible with current HSMs.

Cloning of objects between Luna HSMs requires a shared domain.

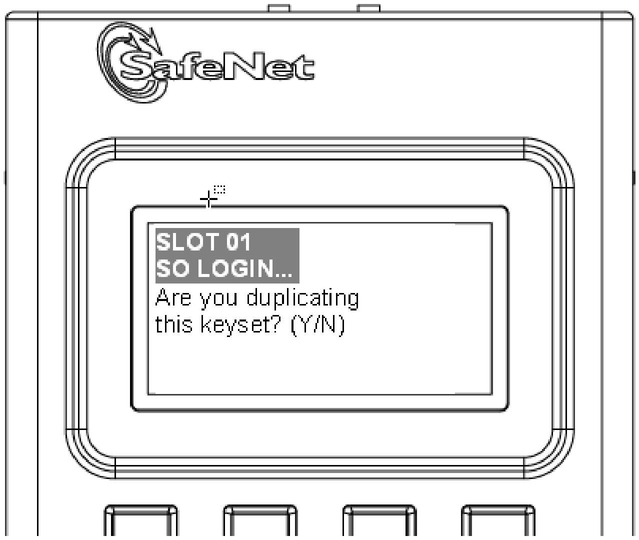
To provide a one-way migration path to move HSM objects from legacy HSMs to modern HSMs, a command `partition setLegacyDomain` allows an old-style domain to be linked to a new-style domain on a K6 or G5 HSM.

Give Me The One-Sentence Summary

If you can account for all the HSMs to which you have presented your red Domain PED Key (meaning that you have maintained strict control of that red PED Key), then you know with certainty that nobody else could possibly have a copy of the sensitive keys that were created on your HSMs or partitions, or cloned to those HSMs or partitions.

Duplicate PED Keys

When you have imprinted any PED Key (having set its parameters: is it re-used; does it have an optional PED PIN, is the secret split into N parts), you are then prompted:



```
SLOT 01
SO LOGIN...
Are you duplicating
this keyset? (Y/N)
```

If you answer YES:

- this invokes the duplication of the PED Key (any number), so that all duplicates can be interchangeable (backups)

- you can now use the original or any of the duplicates to access this HSM or Partition (blue or black keys, respectively), and distribute the others to other personnel or to secure storage
- you should decide how many backup PED Keys are required by your organizational security policies

If you answer NO:

- you are indicating that no duplicates/backups are necessary
- if you eventually require duplicate/backups for your SO PED Keys, you can do so when you initialize another HSM or when you perform an "hsm so-ped-key change"" (saying "NO" to the "reusing" question, and then saying "YES" to the "duplicating" question at that time)
- if you eventually require duplicate/backups for your Partition User/Crypto Officer PED Keys, you can do so when you create another Partition (saying "NO" to the "reusing" question, and then saying "YES" to the "duplicating" question at that time)
- the same possibility is presented whenever you imprint any of the other keys (Domain, RPK, SRK)
- you can also create duplicates of any PED Key, except the purple (SRK), by means of Luna PED's Admin menu.

Multiple or Duplicate PED Keys

The duplicate PED Key option (if you answer "YES" to the Luna PED question "Are you duplicating this PED Key?" during HSM initialization or Partition creation) permits you to issue more than one HSM Admin PED Key (duplicates) and more than one Owner PED Key per HSM Partition, as well as duplicates of any of the other PED Key roles (Domain, Remote PED, SRK, or Audit). The most common use of this feature is to make backups of each PED Key, for secure storage against possible damage to, or loss of, the primary PED Key for an HSM or token.

Your in-house procedures and working arrangements might benefit from having two or more copies of the HSM Admin or Owner PED Keys per HSM. For example, if your procedures require that each work-shift must either sign PED Keys over to the next shift, or sign them into lockup storage, then you need only the single primary PED Key in “circulation”, and you have very secure management of such keys.

However, your procedures could be somewhat less rigid. If it proves more convenient and workable to have each person carry his own PED Key(s) on his person at all times, then a copy will be needed by each person who must ever have access to any given HSM Partition, and to each person with HSM Admin privileges.

In summary, this is an **option**. If you need more copies of a particular PED Key, answer "YES" when you see the "Are you duplicating..." prompt. Any operation that causes Luna PED to offer the "Are you duplicating this PED Key? (YES/NO)" prompt is an opportunity to make as many more copies of that key as you wish. If you already have enough duplicates, just answer "NO" whenever you see the prompt.

The Luna PED (and the attached HSM) do not know how many copies you have made, so you are given the option every time you initialize an HSM or [re-]create a Partition, just in case you might want to create some more duplicates of the currently inserted key. You can also make copies at any time by using the onboard admin menu of the Luna PED 2.x. If your security model allows people to carry PED Keys around, this might be a good argument for imposing the use of PED PIN "something you know" secrets when initializing.

How Many PED Keys do I Need?

You need enough to satisfy your operational and security-policy requirements. How that translates to an actual number of PED Keys depends on your situation. Here is some guidance.

How many off-site full sets do you require? One for many?

Do you intend to use common authentication for many Luna HSMs? The authentication secret on a single blue SO PED Key, for example, could be used with as many HSMs as you like. There is no limit. However if you wish to limit the risk of compromise of a common blue PED Key, you will need to have groups of HSMs with a distinct blue PED Key for each group. [Each time you initialize, the HSM (via the PED) gives you the opportunity to "Reuse an existing keyset" - make the current HSM part of an existing group that is unlocked by an already-imprinted PED Key (or an already-imprinted MofN keyset) - or to use a fresh, unique secret generated by the current HSM.]

How many HSMs per group? One for some?

That will tell you the number of groups, and how many different blue PED Keys you need. Now double that number, at least, to allow for off-premises backup copies to be kept in secure storage in case one is lost or damaged. If you have only one blue PED Key for a group of HSMs, and that PED Key is lost or damaged, the HSMs of that group must be re-initialized (all contents lost) and a new blue PED Key imprinted. In most cases, the contents of an HSM are of some value, so at least one backup per blue PED Key should exist.

One for one?

You (or your organization's security policy) might prefer to have a separate blue SO PED Key - each containing a distinct/unique Security Officer authentication secret - for each HSM in your system. No single blue PED Key can unlock more than one HSM in that scenario. The number of blue keys that you need is the number of HSMs that you have. Now double that number in order to have at least one backup of each blue key.

Many for one?

Does your security policy allow you to trust your personnel? Perhaps you wish to spread the responsibility - and reduce the possibility of unilateral action - by splitting the SO authentication secret, invoking multi-person authentication. Choose the MofN option so that no single blue PED Key is sufficient to unlock an HSM. Two or more blue PED Keys (your choice, up to a maximum of 16 splits of each SO secret) would be needed to access each HSM. Distribute each split to a different person, ensuring that no one person can unlock the HSM.

Having decided that you want (say) three separate people to be present when the SO authenticates to the HSM, you should also allow a few extra splits of that secret, to accommodate accidents, illness, vacations, business travel, or other reasons that would take some key-holders away from the HSM site. Perhaps you settle on two additional splits as sufficient additional key-holders. You have specified M of N equals 3 of 5. Each HSM's SO secret is split into five components, of which any three from that set can combine to reconstitute the SO secret.

Whether you assigned SOs to HSMs on a one-for-one or a group basis (see above), you now multiply that number of SOs by N (the number of splits into which each SO secret is separated). There is no overlap - no split can be part of more than one secret. The number of PED Keys to manage has become significant, especially when you consider that each one (each split of each SO secret) should have at least one backup.

With MofN, you need very good procedures to physically identify and track the various keys.

Partition Black PED Keys

Each HSM has at least one partition. The number depends upon your operational requirement and the number that you purchased, per HSM, up to the product maximum per unit. Each partition requires authentication - a black PED Key.

You have all the same options as described above for the blue SO PED Key(s) - you should have at least one backup per primary black PED Key. That is, you might have multiple partitions, each with a unique authentication secret;

therefore each would have a unique PED Key. Or, you might elect to group your partitions under common ownership, so that groups of partitions (on one or more HSMs) might share black PED Keys.

As with the SO secret, you can also elect to split the partition black PED Key secret by invoking the MofN option [when prompted by the PED for "M value" and "N value" - those prompts do not appear if you chose to "Reuse an existing keyset" at the beginning of the partition creation operation] .

Domain Red PED Keys

Each HSM has a domain. Each HSM partition has a domain. That domain is carried on a red PED Key, and must be shared with another HSM if you wish to clone the HSM content from one to another, for example when making a backup.

Domains must match across partitions for you to clone or back up your partitions, or when assembling HSM partitions into an HA group.

As above, you can make whatever arrangements you wish regarding uniqueness, grouping, MofN (or not), etc., for the red PED Keys.

Other PED Keys

In addition, you might have orange PED Keys if you are using the Remote PED option [orange Remote PED Keys (RPK) containing the Remote PED Vector (RPV)], and you might have purple PED Keys if you are using the Secure Recovery option [purple Secure Recovery Key (SRK) containing the Secure Recovery Vector (the external component of the MTK)], and you might have white PED Keys if you invoke the Audit role Audit role option [white Audit PED Keys containing the authentication for the Auditor, who controls the audit logging feature)] . In any case, you can invoke MofN, or not, as you choose, which affects the number of orange or purple or white PED keys that you must manage.

Orange Remote PED Keys and white Audit PED Keys can be shared/common among multiple HSMs and PED workstations, if desired, just like all other PED Key colors except purple.

SRK secrets are unique per HSM - they are not shared. A purple PED Key is associated with just one HSM. One additional point to remember about purple PED Keys: you can choose to 're-split' the SRK, which places a new secret on a new purple key (or keys if you invoke MofN), but you cannot overwrite a current purple key for the same HSM. The HSM prevents the imprinting/overwriting of any key that carries the current SRV for the current HSM. This is a safety measure.

If you wished, you could overwrite a purple key (or keys) that held an old version of SRV for the current HSM (from a previous re-split). You could also overwrite a purple key that held a current/valid SRV for a different HSM - which would be a problem for that other HSM. The PED and HSM protect the integrity of the current HSM's current SRK, but have no way of knowing whether purple keys from other HSMs are current. In that latter case, you are given the standard PED warning that you are attempting to overwrite a key with data on it (followed by a second reminder "are you sure?"), but you are not prevented from ignoring that warning (twice).

All other PED Key roles allow you to overwrite any key (any color) with a new secret. A warning is given if a key is not blank, but you have the choice to overwrite, or to pause while you find a blank or outdated key ["outdated" in this case means a previously imprinted PED Key that you have made irrelevant by re-initializing an HSM or deleting/re-creating a partition, or other action that makes the secret contained on a particular PED Key no longer relevant; PED Keys do not "age" and become invalid during their service life - only deliberate action on an HSM would cause the secret on a PED Key to become invalid].

With all of the above in mind, it is not possible to suggest one "correct" number of PED Keys for your situation. It depends upon the choices that you make at several stages. In all cases, we repeat the recommendation to have at least one backup in case a PED Key (any color) is lost or damaged.

Using M of N

M of N is designed to provide additional 'eyes' on the setup and deployment of an HSM in a customer environment. The feature implements a balance between this multi-person control and the requirement for these M of N key holders to be present for all operations. The typical deployment of a Luna G5 HSM is either attached to an application server, perhaps to serve as the root of a PKI, or attached to a Luna SA appliance to serve in a similar capacity as part of a "PKI bundle". The typical deployment of a Luna PCI-E (K6) HSM is inside its host or application server, as the root of a PKI, or as the cryptographic engine to an application on that server.

In all those scenarios, it is frequently the case that the HSM and its server(s) are kept in a locked facility and either accessed remotely by secure channels or accessed directly and physically only under specific conditions.

To satisfy these design requirements we have a concept of Partition Activation (see "About Activation " on page 1).This allows administrators of the PED-authenticated HSM to put it into such a state that the calling application is responsible for its own connections and sessions with the HSM, without requiring the presence of the operators for each and every login. This is important when an application or operating system might be rebooted for maintenance, or a power outage might occur (up to two hours duration), and it would be challenging to get the 3 or 5 management personnel together to present the M of N keys. Another way to describe this might be: The black PED Key(s) is presented in order to set the partition into a state of "open for business". When that is true, clients can connect. Clients must still present a challenge secret (previously distributed) to enable them to perform cryptographic operations on the partition. At any time, the holder of the partition User/Owner black PED Keys can close the partition to access (deactivate it) and client applications can no longer access the partition, regardless of their possession of the challenge secret.

A common customer scenario would see the HSM configured and brought into production at a datacenter. This activity would need, first, the quantity M holders of blue SO PED Keys, so that the HSM administrator could log in and create partitions, adjust policies, and so on. Then, quantity M holders of black User PED Keys would be needed in order to activate each partition, making it available for customer connection. At this time the key holders (who would typically be management personnel, rather than day-to-day operational personnel) would give their approval to access the HSM by presenting the M keys at first login, or first partition activation. This is the electronic equivalent of them 'signing off' that the HSM is properly installed where it should be, that the security officer, partition owner and cloning domain holder - as well as the PIN holders if separate - are the correct authorized personnel.

Note that M of N is optional (until you decide to invoke it when a secret is first created), and that it is optional per secret. That is (for example):

- You could choose not to invoke M of N for any HSM authentication secret - so only one blue SO key, and one black User key, one red cloning key, one orange Remote PED key, and one purple Secure Recovery Key, would be needed to access the respective HSM functions. One person could perform each function without oversight.
- You could choose to invoke M of N for some secrets and not for others. For example, HSM-level access could be configured to require multiple blue PED Keys while, say, the partition-level access needs only one black PED Key. The HSM security officer would need M people to agree that she/he had the right to log into the HSM, each time, but any individual partition owner/User could activate her/his own partition with no oversight. The reverse could also be true, with the SO needing just a single blue key for HSM login and HSM administration, but the various partition owners needing multiple persons with black key splits to activate or deactivate their partitions.
- You could invoke M of N for every role, but set different M and N values per role. HSM administration might have a pool (N) of 5 blue keys and need 3 (M) of them for any HSM login event. Meanwhile the pool of black keys (N) for a given partition might be 3 or 6 or 10 or as many as 16, but the number of holders (M) needed to activate the partition might be just 2 (or any number up to N)... and so on, in as many combinations and permutations as make sense for your situation. Similar choices would apply for red, orange, and purple key secrets and for the Audit role. As well, while you can choose to reuse a black PED Key (or an M of N set of black PED Key splits) to create and access multiple HSM Partitions (on a single HSM where permitted, or on different HSMs), you could also choose to imprint

a different black PED Key secret (or separate M of N sets of black PED Key splits) for every partition, or any combination of those options.

Note also that, in addition to the "something you have" authentication factor, each secret-share can also (optionally) have a "something you know" authentication factor. That is, for every split of every HSM secret, you have the option - or not - to declare a PED PIN (see "What is a PED PIN?" on page 57) that must be entered at the keypad when that PED Key is presented.

As with M of N, the PED PIN secret is an option that is chosen via the PED. For each key that is imprinted, you are given the option to set a PED PIN secret (typed on the keypad) in addition to the secret contained inside that PED Key. As each PED Key is unique, it can be given:

- no PED PIN
- the same PED PIN as other members of a set
- a completely different PED PIN.

As you can imagine, combining permutations of M of N with permutations of PED PINs could make for a very complicated security scheme. You have these options; it is up to you to choose and combine them in ways that meet your security needs without over-complicating the lives of your personnel.

M of N General Procedure

Decide whether you want M of N before you initialize your Luna HSM or create a new partition. Read the other pages in this section, to determine what you expect from the feature and whether it fits your policy and operational considerations.

In general, you would determine the number of persons who are to be trusted with single M of N shares and assign that number as N. This applies individually to each authentication secret (blue, black,and red, as well as orange and purple if you are using those). Then, you would decide how many of those people your policy will require to be present whenever the HSM Admin or the Owner logs in to the HSM (or Activates the HSM). Assign that number as M.

You must have quantity N of blank PED Keys in order to implement M of N for a given authentication secret. You need that many keys when you initialize an HSM and choose M value and N value greater than one, when prompted by the PED. They should be blank, or (if they were previously used) no longer needed for any other purpose, because they will be overwritten with new authentication data during this procedure.

To initialize a Luna HSM with M of N (example uses N =5, M = 3) for the SO secret :

1. Open a lunacm session and select the appropriate HSM (if you have more than one installed).
2. Run the `hsm init` command. Type:
   ```
   lunacm:> hsm init -label myLuna
   ```
   The following warning appears:
   ```
   WARNING: Are you sure you wish to re-initialize this HSM?
   All containers [HSM Partitions] and data will be erased.
   Type 'proceed' to delete the container, or 'quit' to exit now
   ```
   Type:
   proceed
3. The initialization proceeds as described in "Initializing an HSM (Trusted Path option)" as each secret is created, the PED prompts for "M value =" and "N value =". For any secret, set M and N equal to "1" if you do not wish to invoke MofN secret splitting for that secret. Set M and N larger than "1" if you require secret splitting (multi-person access control) for that secret - SO, domain, etc.

4. If you are splitting the current secret, insert a blank PED Key and press [Enter] on the PED touchpad. Create a PED PIN for that split, if you wish, or just press [Enter][Enter] for no PED PIN.
5. The same sequence of PED prompts reappears. Insert another same-color PED Key and press [Enter]. During this process, you must supply a fresh same-color PED Key at each prompt. Do not present the same PED Key twice.
6. Repeat until Luna PED stops asking for more that-color PED Keys. It requests as many blank keys as the quantity "N value" that you supplied in the opening PED prompts for this secret's creation (in this example, 5). Label the PED Keys to avoid mix-ups.
7. The procedure that you started in "Initializing an HSM (PED Authenticated option)" continues until you need to login. First, Luna PED prompts for the blue HSM Admin PED Key, as described in the standard procedure. Then, Luna PED begins demanding blue imprinted PED Keys until it has received "M" different ones from that MofN set.
8. Continue to insert one of the imprinted SO-split (blue) PED Keys and press [Enter], and repeat until Luna PED is satisfied. You must insert M different keys when authenticating. Luna PED recognizes if you attempt to present the same key more than once during an authentication attempt. So, in this example, 3 of the 5 blue keys are needed - any 3 from that imprinted set of 5, as long as they are different.
9. Complete the rest of the hsm init procedure, then hand out the imprinted PED Keys to separate, trusted persons whose job it is to come together whenever HSM authentication is needed.

Once you have completed the initialization with M of N, then every time you need to authenticate to this Luna HSM, you will need M of the N blue or red (or any other authentication - black, orange - for which you specified M greater than 1) PED Keys in that group.

Normally, M should be smaller than N, so that you can login to the HSM (or Activate/autoActivate it) while some of the trusted N persons are away for business, vacation, illness, etc.

How to Add an M of N Requirement Where There Was No M of N Before

Historical Note:

On Luna SA 4.x systems, if one HSM had M of N (using the legacy green PED Keys), and you wanted another HSM to use the same M of N splits, you had the option to clone M of N from the first HSM to the second.

Current Practice:

With Luna HSM 5 where M of N is a condition of each authentication secret independently, there is no provision to "clone M of N". Instead, if you wish to have two HSMs share the same M of N scheme, you must initialize one with the desired scheme, then initialize the second (and any additional) HSM and have it re-use the secret(s) from the first HSM.

At secret-creation time for the HSM, when the PED is invoked, the PED asks if you wish to re-use an existing secret. If you say "yes" to that question, then the PED expects you to offer a PED Key (for example a blue PED Key, when you are initializing) that is already imprinted with a suitable secret. If you offer a blue key that contains a partial secret - a split from your other HSM - the PED accepts that key. The connected HSM recognizes that the secret is only a split, not a full SO secret, so the PED demands additional keys from that set, until it has received M of them, enough to reconstitute the secret. It will not accept fewer than M different portions of the secret, and it will not accept members of another set.

Once the reconstituted secret has been imprinted on the new HSM, then that HSM can accept any M splits out of the full set of N, even though it has never seen some of those splits. Both HSMs now accept the same M of N authentication secret. You can do the same, individually for any of the other secrets on the new HSM (black partition User keys, red cloning Domain keys, orange RPV keys). The only exception is the purple PED Key (or Keys), since the MTK and SRK are unique to each HSM and cannot be cloned or shared.

Purple Keys:

You *can* duplicate a purple PED Key while you are in the process of imprinting it (SRK enable, SRK resplit).

You *can* split the purple-key secret (which is already one split of a larger secret inside the HSM), via M of N, so that the Secure Recovery Vector secret needs multiple purple key holders to invoke it.

You *can* re-split the internal MTK of your HSM, resulting in a new SRV portion imprinted on the external purple key (or keys, if M and N are greater than 1).

You *cannot* generate a new master secret on the HSM - the MTK is unique and permanent for each HSM and can be changed only by remanufacturing. Factory reset and initialization have no effect on the MTK.

You *cannot* imprint a purple key secret from one HSM onto another (for the same reason as above), unlike all the other key colors where sharing/grouping are important options.

You *cannot* duplicate a purple PED Key via the PED's stand-alone (no HSM present) Admin menu. The "raw" duplication function, which works for all other PED Keys, refuses to duplicate purple keys. This is a security feature, so that no one can duplicate a purple key without access to the HSM that created it. This applies to splits of the SRK as it applies to a single SRK purple key.

Implementation Suggestions

Here is one suggestion for having the security benefit of M of N, including backups, but without the risk of accidentally mixing members of original split set and backup split sets. [Remember, the risk is not that members of "original" and copy sets can't work together - they do - the risk is accidentally having copies of the same key together. The PED requires different splits when combining quantity M splits to recreate the authentication secret. If you offer it one split and then a copy of the same split (because they all look alike and you accidentally gathered them into incorrect groups), the PED will reject the identical offering because it assumes you are offering the same split twice.]

If your M and N numbers are small, like (say) 3 of 5, simply declare a large N (up to 16 splits is permitted, so in this case use 3 of 15) and simply gather them into groups of (say) 5, one group for regular operations, one group for standby, one group for off-site backup storage. In this way, all the splits are valid together in any combination- any three of the 15 can unlock the HSM. You do, of course, need to control distribution of, and access to, all those secret-split keys.

If your M number is larger, then this idea becomes less practical, since you have a maximum N of 16 to work with. It depends on how many sets of M you need. At the very least, you should have one backup of every HSM authentication secret, preferably in secure off-site storage.

M of N is not for everybody. For those who need it, it is crucial, and the added administrative task is a "cost of doing business". If you don't need M of N in your security regime, then we suggest that you not use it.

If your security policy demands that you use M of N multi-person access control and also demands that M be relatively large, consider carefully if your policy might need review. Any security regime should be no more complicated than it needs to be - no more complicated than yields a net-positive security benefit. The more complicated or onerous a security policy, the more your own personnel - even the most trust-worthy - are motivated to circumvent or simplify, in order to get on with their tasks.

Complexity When Managing PED Keys

The options to create group PED Keys and duplicate PED Keys can introduce complexity and another kind of risk to the management of PED Keys, especially when the options are combined. In many establishments, security policy demands that passwords be changed on a regular basis. Naturally, passwords/PINs on HSMs, Partitions and tokens and PED Keys can be changed as needed.

However, what might be a simple procedure for a single key (Change PIN) can quickly take on new dimensions when there might also be a backup PED Key in off-site safe storage, and there might be several working copies of the PED

Key in the hands of Owners, alternate/backup Owners, and alternate/backup HSM Admins. Additionally, there might be several tokens, Partitions, or HSM Servers that are unlocked by any of the PED Keys (if you chose the group PED Key option when creating any of those).

The issue is that when authentication data is changed on a PED Key, it must be changed on the associated HSM or Backup Token at the same time; otherwise the two no longer match and the PED Key can no longer unlock the HSM (or Partition) or the token. The changePw procedure does take care of this for the HSM (or Partition) or token and for the blue (or black, as appropriate) PED Key that is in the Luna PED slot when the change command is issued. There is also provision (explained in following pages) for having other accessible PED Keys updated, during the procedure, to maintain synchronization with the HSM (or Partition) or Backup Token.

But, what about the set of backup PED Keys that you have sensibly stored off-site? If they are not brought in and updated during the same update procedure, they are no longer backups. Your security and maintenance procedures must address this situation.

To ease the task of updating multiple PED Keys,without a complicated dance involving all sets during the same update event, the PED provides a method of stand-alone, "raw" key duplication.

[<] to exit Local PED mode to the main PED menu
[4] to enter Admin mode
[1] to enter PED Key mode
[1] again to bypass key login, which is not applicable to the iKey 1000 model in current use
and then
[7] to duplicate whichever key is presented next.

This is applicable to all imprinted PED Keys except the purple SRK (excluded for security reasons).

Because the above is a "raw" duplication, there is no opportunity to modify any PED PIN that is already associated with the presented source key. Duplicates by this method are exact.

Once you have updated any or all members of a working set of PED Keys, you can take one of those keys and a PED to any other location where duplicate sets were maintained (onsite backup, offsite backup, etc.) and update your backups without any need to involve the HSM. Always be aware of the location and state of any Luna PED key, and keep scrupulous records of all changes and hand-offs. Your security auditors will thank you.

General Advice on PED Key Handling

In addition to the cardinal admonitions about careful physical security and prompt, thorough backups of your HSM partitions and PED Keys, here are some practical tips to make the tasks as easy as possible.

Keep a Log

Keep careful records, both of the regular backup procedures, and of who has possession of any token and any PED Key at any time. Your records should show every hand-off or change of possession and your policy should enforce it. Proper security protocols demand that you be able to account for all primary devices (HSM Servers, tokens and PED Keys) at all times, without exception. Establish strict procedures governing when and how those devices may enter storage, be removed from storage, or change hands among users.

When performing backups and other maintenance functions (such as changing PINs on keys and HSMs), log the event, but also keep a worksheet of notes so that if the task is interrupted you can resume it without confusion or hesitancy as to which devices have been altered and which have not. To help in that regard, see the next section.

Apply Meaningful Labels

This suggestion has two aspects relating to everyday handling convenience and to the previous section, “Keep a Log”:

1. Apply text-string labels to your HSM Servers and tokens.
2. Apply physical labels to the exterior of the physical devices.

In the first case, a unique, easily identifiable word or phrase serves as a final check in `lunacm` or at the client when you are about to perform an action that could alter an HSM or its contents. You might consider a label consisting of a part (perhaps a word) that identifies the domain to which the HSM belongs, and another part (perhaps another word or a number) that identifies it as a particular member of that group.

The second case, physical labels, applies to HSMs and PED Keys.

When handling multiple HSMs and keys, it is easily possible to become confused as to which ones have been updated and which ones are yet to be updated. Worse (if you are using common administrative group PED Keys) would be restoring onto the wrong Partition or HSM, from a backup.

General physical handling is made easier if you have a way to identify a device visually. Easy identification facilitates log-keeping.

Do not cover or obstruct the connector end of a PED Key.

Keys

PED Keys have different roles. Colors help to easily distinguish the roles and you should use the labels included with the product (blue, red, black, orange, white, and purple) to mark PED Keys before you initialize them. The additional suggestions on this page are about applying *additional* labels (stickers, tags, other) of your own, to identify specific keys and key sets and where they fit in your operational scheme.

The PED Keys might further be in need of visual identifiers if you elect the M of N option, which adds several, visually-similar keys to the mix. It might be useful to identify the following:

- Which keys (blue, black, red, orange, white, purple) are associated with which HSMs or Partitions).
- Which black keys are associated with which Partition and client. It normally makes sense to associate a key to a title or function, rather than to a specific person.
- Which key is which in an M of N group. This is particularly useful when the SO is initializing HSMs and keys (and could be accomplished by temporary labels in that situation).

You must decide whether visual identifiers of M of N status of keys would be useful once the keys and HSMs are in operation (or in backup safe storage), or whether your security requirements would prohibit such tags or markings.

Updating PED Key for a Backup

There is no explicit provision for changing the authentication for a Backup HSM. If you need to have new authentication for your Backup HSMs, then perform a new Backup operation.

Performing an HSM Backup or a Partition Backup will initialize the HSM and allow you either

- to imprint a new authentication secret(say "NO" to the "reuse ID" question, which causes a new random secret to be created and imprinted on both the PED Key and the Backup HSM) ,

or else

- to share the authentication secret(say "YES" to the "reuse ID" question, which takes the token authentication from the PED Key that you insert, and not the other way around) that is already in use on other HSMs.

Updating PED Keys – Example

The following is just an illustrative example of changing PED Keys (or the authentication secrets on the PED Keys and the corresponding secrets on HSMs). For the purposes of the example, we will ignore additional complicating factors like PED PINs and M of N that might apply to your situation.

Say, for example, that you had shared PED Keys among three HSMs, and that you also made three other copies of that SO PED Key, so that you and two other persons could each work with one (or any) of the HSMs, and so that the fourth PED Key could be stored away securely.

Risk of Losing access

If you were to “Change PIN” for your own PED Key (and your HSM), then that PED Key would work for that HSM, but the PED Key would no longer work for any of the other HSMs and none of the other PED Key holders of your group could access your HSM. Your HSM would expect the new PIN, and the other people would be holding PED Keys with the original PIN.

Immediately, you see that any time you change passwords (PINs) it must be done for all HSMs (or Partitions) in such a group, and for all PED Key duplicates associated with that group of HSMs (or Partitions if you are changing black User PED Keys).

PIN-change Procedure for Multiple HSMs

⚠ **CAUTION:** You must retain at least one old-PIN PED Key until all HSMs have the new PIN, or you will find yourself unable to access old-PIN HSMs.

1. Choose an HSM and login as SO (with a blue PED Key).
2. Request a change of SO PED Key:

   ```
   lunash:> hsm changePw
   ```

3. Respond to the PED prompts as follows:

   ```
   Getting current SO PIN...
   Reading SO PIN...
   Insert a blue Key
   ```

   This is where you insert a currently valid SO PED Key to confirm that you are the key holder.

   ```
   <Press ENT>
   ```

   The PED requests the key because an indeterminate amount of time might have elapsed since the last HSM login and confirmation is needed that the person asking for a change of secret is the person who logged in (and not an unauthorized person taking advantage of an unattended login session).

   ```
   Reading SO PIN
   Please wait..
   Would you like to reuse an existing keyset? (Y/N)
   ```

   Here you respond "NO" so that a new SO secret is generated.

   ```
   M value  (1-16)
   >0
   M value  (1-16)
   >0
   ```

```
Writing SO PIN...
Insert an SO Key
```

This is where you insert the first SO PED Key to be overwritten; it might be the same one that you just inserted to authenticate as SO

```
<Press ENT>
Writing SO PIN...
PED Key will be overwritten
```

The PED detects existing (old) data on the key and warns you that it will be overwritten if you proceed.

```
<Press ENT>
Writing SO PIN...
Enter new PED PIN
```

This is a new secret, so you have the opportunity to add a PED PIN to it, if you wish.

```
Writing PED PIN...
Confirm new PED PIN
Are you duplicating this keyset? (Y/N)
```

Answer "YES" because you want to overwrite the old secret on two of the remaining three PED Keys (in this example).

```
Writing SO PIN...
Insert SO key
```

This is where you insert the second SO PED Key

```
<Press ENT>
Writing SO PIN...
PED Key will be overwritten.
<Press ENT>
Writing SO PIN...
Enter new PED PIN
```

You can add a PED PIN to this duplicate key if you wish, or not. If you add a PED PIN it does not need to be the same as on the other key.

```
Writing PED PIN...
Confirm new PED PIN
Would you like to
make another'
duplicate set? (Y/N)
```

Respond "YES" and make the change on the third SO key, but leave the fourth key with the old secret for now.

```
Command Result : 0 (Success)
[luna22] lunash:>
```

At this point, you now have ONE HSM and three of your four SO keys imprinted with the new SO authentication secret. Ensure that you keep the keys separate and well identified. One PED key MUST retain the old secret until all HSMs are updated to the new secret.

4. Go to the second of your Luna appliances, login as admin.
5. Request a change of SO PED Key (this time you will not be changing key contents, you will be logging in with the old secret, then copying the new secret from one of the updated keys onto the second HSM):

   ```
   lunash:> hsm changePw
   ```

6. Respond to the PED prompts as follows:

```
SO login...
```

This example step shows that if you had not already logged in prior to requesting "hsm changePw" then a login is forced.

```
Insert blue PED Key
```

Insert the old-secret PED Key, to login -- this HSM still has the old secret.

```
<Press ENT>
Getting current SO PIN...
Reading SO PIN...
Insert a blue PED key
```

The system does not track how long ago the login occurred, so before a key change is permitted, it requires you to prove that you are the valid keyholder, by producing the key again.

```
<Press ENT>
Reading SO PIN
Please wait...
Setting SO PIN
Would you like to
reuse an existing
keyset? (Y/N)
```

Here you respond "YES" so that the new SO secret will be read from the new-secret-containing key that you are about to insert.

```
Reading SO PIN...
Insert a blue PED Key
```

This is where you insert a new-secret SO PED Key so that its secret can be read and then imprinted on this second HSM.

```
<Press ENT>
Would you like to
make another'
duplicate set? (Y/N)
```

Respond "NO". This HSM now has the new secret.

```
Command Result : 0 (Success)
[luna22] lunash:>
```

At this point, you now have TWO HSMs and three of your four SO keys imprinted with the new SO authentication secret. Ensure that you keep the keys separate and well identified. One PED key MUST retain the old secret until all HSMs are updated to the new secret.

7. Remove the new-secret key from the PED and place it with the other new-secret keys.
8. Bring a PED and the remaining old-secret key to the third appliance and login as admin.
9. Request a change of SO PED Key (you will be logging in with the old secret, then copying the new secret from one of the updated keys onto the third HSM, then overwriting the final old-secret key with the new secret, once the old secret is no longer needed).

> **Note:** You can explicitly login (with "hsm login") before issuing "hsm changePw", or you can wait until you issue the change command and be prompted to login.

```
lunash:> hsm changePw
```

10. Respond to the PED prompts as follows:

```
SO login...
Insert blue PED Key
```

This prompt appears if the HSM was not already in the login state. Insert the old-secret PED Key, to login -- this HSM still has the old secret.

```
<Press ENT>
Getting current SO PIN...
Reading SO PIN...
Insert a blue PED Key
```

Here, the PED wants the same secret that you used to login.

```
<Press ENT>
Reading SO PIN
Please wait...
Setting SO PIN
Would you like to
reuse an existing
keyset? (Y/N)
```

Here you respond "YES" so that the new SO secret will be read from the new-secret-containing key that you are about to insert.

```
Reading SO PIN...
Insert a blue PED Key
```

This is where you insert a new-secret SO PED Key so that its secret can be read and then imprinted on this third HSM.

```
<Press ENT>
Would you like to
make another'
duplicate set? (Y/N)
```

Respond "YES", and supply the last old-secret PED Key as the "blank".

```
Command Result : 0 (Success)
[luna22] lunash:>
```

At this point, you now have all three HSMs and all four SO keys imprinted with the new SO authentication secret.

If you prefer to be more cautious, you could have left the final PED Key with the old secret until you verified that all three HSMs are now unlockable by the new secret, and only then invoke the command one more time to imprint the last key with the new secret.

Alternatively, on a Luna PED 2.x, you can perform iKey PED Key copying or duplication at the PED without invoking commands at the HSM (however you still require a connection between PED and HSM to power the PED).

Note: You can perform the same operations with blue SO PED Keys, in similar circumstances, and observing the same precautions. Also, this sort of operation could be scaled up for larger groups of HSMs (if they share a group-User or group-SO PED Key) and for larger numbers of duplicate PED Keys.

Note: To avoid confusion, it's probably best if you mark each key to identify it, and keep a careful log of which key and which HSM has what operation done to it, at each step.

Frequently Asked Questions

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

How should Luna PED Keys(*) be stored? (*Model iKey 1000 for use with Luna PED2)

Physically, they are electronic devices, and should be stored in environments that are not subjected to extremes of temperature, humidity, dust, or vibration.

With that said, PED Keys that have their protective connector-caps in place are quite robust. PED Keys that have their caps on when not immediately in use have survived years of daily use being carried around in office-workers' pockets, here at SafeNet's Luna labs.

Procedurally, they should be labeled and stored (filed) so that they are readily identifiable according to the HSM(s), the partitions, and the roles with which they have been associated.

So I shouldn't keep all the PED Keys for all my Luna HSMs in one box in a desk drawer?

No. The only place where that might be appropriate is in a test lab where HSMs are constantly re-configured for test purposes, and where they never contain important cryptographic material. PED Keys are just generic iKeys until you make them into specific kinds of PED Key by your administrative actions with Luna HSMs and Luna PED. Once a blank iKey has been turned into a Security Officer (blue), Domain (red), Partition Owner/Partition User (black), Audit (white), Secure Recovery (purple), or Remote PED (orange) key, you must ensure that it is labeled as such and that you handle it and store it in a way that it can never be mistaken for a different PED Key.

We have had at least one customer call us in a panic because they had "lost" the SO and Domain and Partition keys to an enterprise-critical HSM. They actually still had those keys, mixed with many others in a box. As you know, Luna HSM authentications do not permit a lot of "guessing". For example, you get only three tries to present the correct blue PED Key for HSM SO login, before the HSM contents are lost forever. A customer staffer, new to her job asked: "You make the HSMs and keys, why can't you just give us another one?" We had to explain that there is no 'back door', ever, to a Luna HSM. We (SafeNet) did not make her PED Keys. We made the iKeys, and her predecessor created them as the PED Keys for her organization's HSMs.

Fortunately, that customer's critical HSMs were in an HA configuration that had not yet synchronized, and the secondary HSM was still in logged-in state. After trying several red PED Keys it was possible to get a backup of the secondary HSM and restore onto a re-initialized primary HSM. After that, our responding support engineer spent many hours teaching the customer staffers the basic security and HSM administrative knowledge that had been lost due to staff turn-over at that company. That enterprise customer has since installed rigorous procedures and documentation for handling of HSMs and HSM authentication secrets.

I've lost my purple PED Key. Or, I forgot my PED PIN for my purple PED Key.

You are likely in for some cost and disruption, but this is not necessarily a fatal mistake.

At the present time (this note is written in February 2013) there is no way to recover from a tamper or from Secure Transport Mode if the external split of the Master Tamper Key (the SRK) is not available. If you haven't got a backup purple key, your HSM is locked the moment it experiences a tamper event, or if it was placed in Secure Transport

Mode. The same applies if you do have the key, but have forgotten/lost the numeric PED PIN that you applied when the purple key was imprinted with the Secure Recovery Vector (the external split of the MTK). Either way, you must obtain an RMA and return the HSM to SafeNet for remanufacture. All HSM contents are lost.

As with every PED Key that you imprint, we recommend that you make at least one backup copy of the purple PED Key, as well. If you can find that valid backup purple key, you can recover the HSM and make a new split, without problem. If the purple key that you lost was the only one... then see the preceding paragraph.

Note that simply not having the external MTK split available is not the end of your HSM and its contents. As long as it has not been tampered, or was not placed into Secure Transport Mode, then the HSM is still working and is perfectly accessible to other key-holders. However, you should immediately back-up all important HSM contents to other HSMs and have SafeNet remanufacture the affected HSM. When that HSM is returned to you, it will be in one of two states:

a) it will have both MTK splits internal (no SRK created), or

b) it will have a new MTK and a new SRK (purple PED Key) if you requested that we ship the HSM to you in Secure Transport Mode.

In the first case, you have a "new" working HSM and can decide what you wish to do with respect to SRK - if it is not necessary to your security regime, simply never declare an external split and you will never need to worry about purple keys. Tamper events (if any) will be logged, but will recover automatically when the HSM restarts.

In the second case, you receive the HSM back from SafeNet, in STM (as requested) and you receive the associated purple key (SRK) by separate courier. You recover the HSM from Secure Transport Mode. At that point, you can elect to disable SRK (return the external split inside the HSM, simultaneously generating a new internal split pair, and invalidating your purple key). OR, you can elect to make a new external split. This imprints a new purple key (SRK) and invalidates the one that we shipped to you. You should make at least one backup copy of the new purple key when it is created, and take better care of your imprinted PED Keys in future.

Also, if your security regime does not require multi-factor authentication, then see the next question, about PED PINs.

Do we really need to include a PED PIN with each PED Key?

Not at all. Or, rather, you do if you already set a PED PIN when you initialized/imprinted that PED Key. But a PED PIN is an optional item when you first initialize an HSM or create a partition, etc. You have the choice, and you don't want to impose a PED PIN requirement on yourself without good reason.

A PED Key is single-factor physical authentication - "something you have". If that is sufficient to satisfy your organization's security requirements, then you do not need to impose PED PINs.

You can just press [Enter] on the PED keypad when the PED Keys are being imprinted (that is just press the [Enter] key with no digits), and you would never be troubled by a PED prompt about PED PINs again.

PED PINs are an option - until one is imposed; then it becomes mandatory. Only if your security regime requires two-factor authentication should you consider applying PED PINs to your various PED Keys. Where the physical PED Key is "something you have", the PED PIN is the second factor, the "something you know". A PED PIN is a convenient and effective second factor, but it does represent an additional item for you to remember and to track.

If you lose track - if you fail to remember a PED PIN, or if you have several and don't remember which is which - you can find yourself locked out of your HSM or your HSM partition as surely as if you lost the physical PED Key. More surely, in fact, since you probably have physical backups of your PED Keys (you do, don't you?). Remember, typing a wrong PED PIN on the PED's keypad is the same as offering the wrong physical PED Key to the HSM. It counts as a bad login attempt. PED PINs are good and essential when you need one, but they are not something to impose without a solid security-based requirement.

CHAPTER 10
Performance

This chapter describes the actions you can take to maximize the performance of your HSMs. It contains the following sections:

HSM Information Monitor

An HSM administrator might find it helpful to know how busy the HSM is, currently, 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 number. The HSM firmware tracks the overall time elapsed since the last reset (Up-Time), and the overall time during which the processor was not performing useful work (Idle-Time).

On request, the HSM calculates "Busy-time" over an interval, by subtracting Idle-time for that interval from Up-time for the interval. Then, the load on the processor is calculated as the Busy-time divided by the Up-time, and expressed as a percentage.

You can use the available commands for a single, one-off query, which actually takes an initial reading and then another, five seconds later (the default setting), in order to calculate and show the one-time difference.

You can specify a sampling interval (five seconds is the shortest) and a number of repetitions for an extended view of processor activity/resource usage. The resulting records, showing the time of each measurement, the percentage value at that time, and the difference from the previous measurement, can be output to a file that you import into other tools to analyze and graph the trends.

By watching trends and correlating with what your application is doing, you can

- determine the kinds of loads you are placing on the HSM,
- seek efficiencies in how your applications are coded and configured,
- plan for expansion or upgrades of your existing HSM infrastructure,
- plan for upgrades of electrical capacity and HVAC capacity.

Notes about Monitor/Counter Behavior

When performing certain operations the HSM reaches its maximum performance capability before the counter reaches 100%. This occurs because the counter measures the load on the HSM’s CPU and the CPU is able to saturate the asymmetric engines and still have capacity to perform other actions.

Also, symmetric cryptographic operations cause the counter to quickly rise to 90% even though there is significant remaining capacity. This behavior occurs because, as the HSM receives more concurrent symmetric commands, its

CPU is able to handle them more efficiently (by performing them in bulk) – thus achieving more throughput from the same number of CPU cycles.

See "hsm monitor" on page 1 of the *Lunacm Command Reference Guide* for more information.

Timeouts

As a general rule, do not adjust timeout settings (either via the interface or in config files) unless instructed to do so by SafeNet Customer support.

Changing some settings can appear to improve performance until a situation is encountered where a process does not have time to complete due to a shortened timeout value.

Making timeouts too long will usually not cause errors, but can cause apparent performance degradation in some situations (HA).

Default settings have been chosen with some care, and should not be modified without good reason and full knowledge of the consequences.

If adjusting the configuration files for any reason:

Note: NEVER insert TAB characters into the chrystoki.ini (Windows) or crystoki.conf (UNIX) file.

Generating Large Keys

Luna G5 can create and manipulate cryptographic keys as specified by you (depending on type of key, etc.). However, for optimum performance we use default settings that assume a common range of sizes.

For example, the majority of our customers are expected to be using (say) RSA 1024-bit and 2048-bit keys. The nature of the RSA keygen math is such that multiple attempts to create a key might fail (testing for primality and other characteristics of intermediate numbers that are created and used in the process). The firmware and software take care of counts and retries when a keygen operation is requested. The only configurable timeout is for the driver, and that is set in the Chrystoki.conf (or chrystoki.ini) file.

Large Keys Need Longer Timeouts

The default setting for "KeypairGenTimeOut=600000" is appropriate for keys up to 2048 bits in size. For larger keys, the process might take longer and result in a timeout before completion. Therefore, if you intend to create 4096-bit RSA keys (or larger), we suggest that you explicitly set "KeypairGenTimeOut=2700000" in the "Luna" section of the chrystoki.conf or chrystoki.ini file.

CHAPTER 11

Remote PED

This chapter describes how to use the remote PED to authenticate to an PED-authenticated HSM at a remote location. It includes the following sections:

About Remote PED

The Remote PED concept (Luna PED with Remote Capability) was introduced to satisfy a need to administer HSMs that are housed away from their owners/administrators, at physically remote sites or inside heavily-secured premises, where obtaining local physical access to the HSM is difficult or time-consuming.

Remote PED provides administrative convenience similar to remotely accessing a Password-authenticated HSM, but with the added security and role separation of PED authentication. The remote system is asked to perform an HSM function (this is the Administration aspect); it demands the relevant PED Key (the Authentication). With local Luna PED this would mean that someone standing beside the remote appliance would need to connect a Luna PED, insert the requested PED Key and press [ENTER].

Remote PED provides a means to perform sensitive operations on HSMs that have access secured by Trusted Path (PED) Authentication, without being physically present to insert PED Keys and press PED buttons on a Luna PED connected directly to the HSM.

The feature requires:

- a Remote PED Server on a workstation that connects over a secure network link to
- a Remote PED Client in the computer or appliance that contains the HSM, and
- a SafeNet Luna SA PED 2.5.0-2 or greater, **with** the Remote PED feature installed, which has the capability to operate in Local PED or Remote PED mode, as needed; not every PED 2.5.0 includes the Remote PED feature - that PED capability must be ordered specifically and factory installed, and
- an orange RemotePED PED Key, which provides the authentication for the Remote PED connection between the workstation computer (with Luna SA PED 2 connected and PEDServer running) and the remotely located Luna SA appliance with the RemotePED client running.

Term	Meaning
Remote PED	A Luna PED, with Remote capability, connected, powered on, and set to Remote mode.

Term	Meaning
RPV	Remote PED Vector - a randomly generated, encrypted value used to authenticate between a Remote PED (via PedServer) and a distant Luna HSM (PED Client).
RPK	Remote PED Key - an orange PED Key, the repository of an RPV value, for use in the Remote PED process.
PedServer	The PED server program that resides on a workstation and mediates between a locally-connected Remote PED and a distant PEDClient (running at a distant Luna HSM).
PEDClient	The PED Client program - embedded in the case of a Luna appliance, or installed on a computer with a contained Luna K-card HSM or with a USB-connected Luna G5 (or Backup) HSM - anchors the HSM end of the Remote PED service and initiates the contact with a PedServer instance, on behalf of its HSM.

Why do I want it?

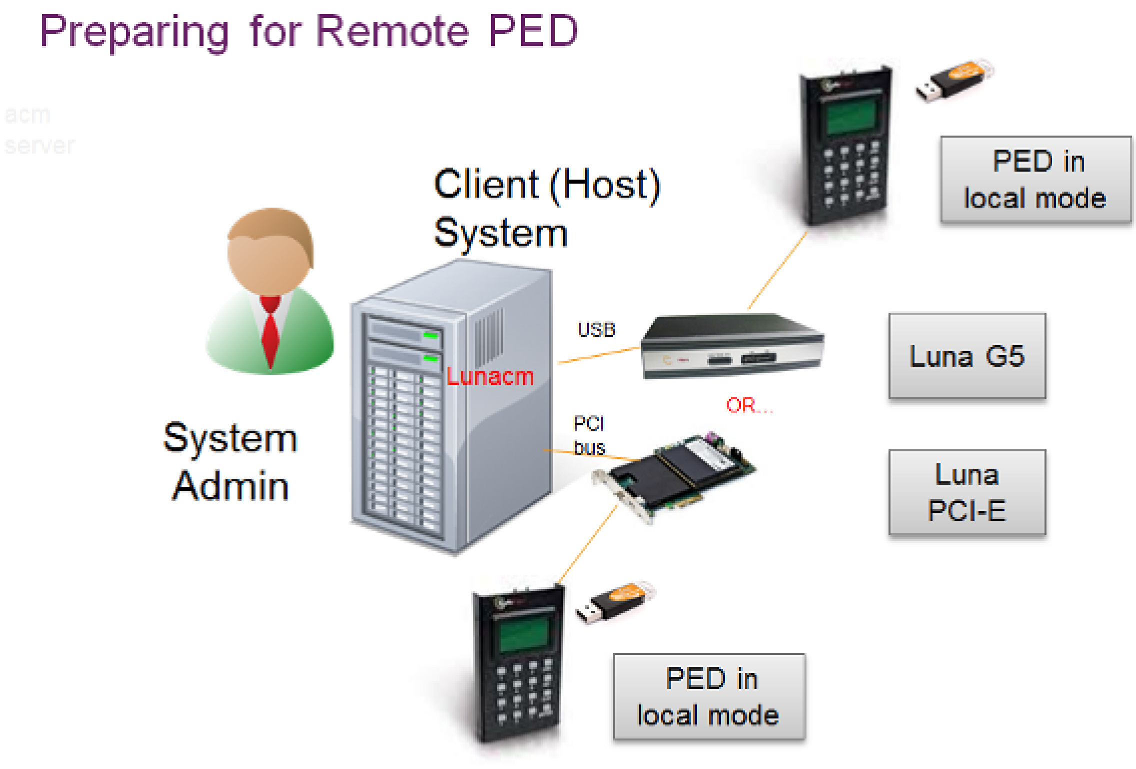
You want to locate your operational appliances at remote locations or multiple locations around the city, country, world, and be able to administer them fully, from one location, without need for site visits and without carrying of PED Keys through unsecured areas.

How does it work?

The HSM must initially be configured with a local PED, in order to set its authentication and create a relationship between the HSM and an orange PED Key (RPV, or Remote PED Vector). That RPV, carried via the orange PED Key (RPK), is the means by which a PED at a remote (PedServer) location can be recognized and trusted over a distance, by an HSM that shares the same RPV.

During the imprinting process, the HSM can take on the RPV of an existing orange PED Key (RPK, or Remote PED Key), or the HSM can generate a new RPV and imprint it on an orange PED Key.

The diagram shows the preliminary imprinting step, where the HSM and (at least one) orange PED Key are made to share an RPV. Again, this must take place via a Luna PED that is connected directly to the HSM. The administrator could be co-located with the HSM, or could be elsewhere issuing the commands, but either the administrator or an assistant must be present at the HSM to present the orange PED Key for the RPV imprinting. Once that is completed, further PED operations can be untethered from direct local PED connection and moved anywhere, along with that RPV-bearing orange PED Key.

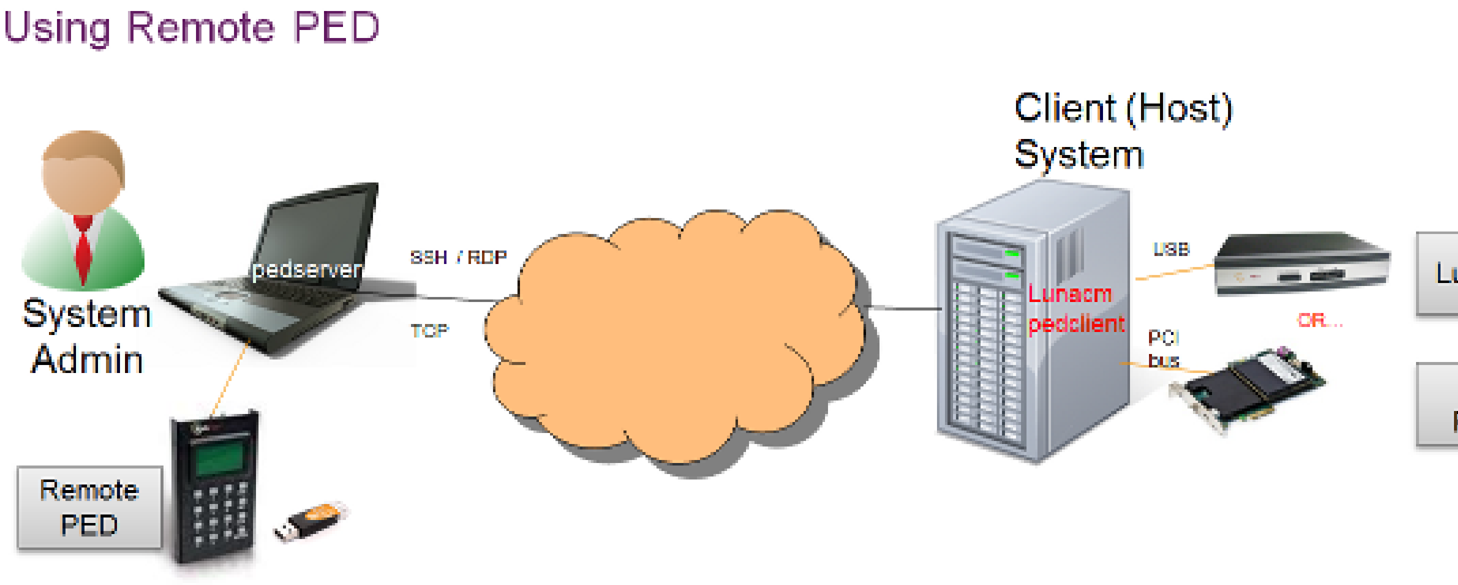


The HSM is then shipped and installed at its remote location.

At your administrative location, a workstation is configured with special (PedServer) software, and a Luna PED 2 Remote (remote-capable PED) is connected via USB to that workstation.

Using SSH, you open an administrative session (connect and log in as "admin") on the remote HSM. You launch pedClient on the HSM host, and tell the HSM to expect a remote PED, rather than local PED. You issue commands as needed.

When an HSM command requires authentication to the HSM, the HSM looks for a remote PED server with the same Remote PED Vector. If it can authenticate properly with that remote PED server, the HSM accepts authentication data via that connection.



One-to-One Remote PED Connections

A SafeNet Luna HSM on a host that is running pedClient can establish a Remote PED connection with any workstation that

- is running PEDserver.exe,
- has a suitable Remote PED connected (version 2.5.0-2 or later), and
- has the correct PED Keys (including the orange key) for that HSM.

However, the Luna HSM can make only a single connection for Remote PED operation at one time. The current session must timeout or be deliberately stopped before another workstation can be called into a Remote PED connection with that Luna HSM.

Similarly, a given workstation can enter into a Remote PED connection with any Luna HSM with PEDClient that initiates such a connection (provided the proper PED, PED Keys, software, etc. are all in place), but it can make only one such connection at a time. This contrasts with SSH connections, where that same workstation could have multiple SSH windows open to multiple admin sessions on a single or multiple Luna HSM host.

There is no requirement for the workstation providing the Remote PED connection to be the same one providing the SSH session to the HSM host admin, nor is there any requirement that they be different workstations.

Priority and Lockout

A Remote PED connection is always initiated from the Luna HSM - a workstation cannot invoke a Remote PED session as a Remote PED function. That is, you could be sitting at Workstation "A", with a command-line window open, in which you can run the PedServer.exe, and there is no provision to use that program to connect to the Remote PED client on a Luna HSM-attached computer, or a Luna SA appliance. Nevertheless, you could open an SSH window on that same workstation "A" (or on any other computer), connect to the Luna HSM's host, log in, and tell the HSM to initiate a Remote PED connection (ped connect) with workstation "A". The two functions (a communication connection for Luna shell [lush] and a communication connection for Remote PED operation) are completely separate.

When a Remote PED connection is in force, the local PED interface to the HSM is disabled. If a local PED operation is in progress, it is not possible to start a Remote PED connection until the current local-PED-mediated HSM operation completes. But it must be an active operation sequence - merely having a local PED connected to the HSM does not lock out the initiation of a Remote PED connection. For example, if you had started an HSM command that began using a connected local PED and PED Key for authentication, AND you started an SSH session in which you issued the ped connect command, one of two things would happen:

- the ped connectcommand would begin executing, would pause while the local-PED operation (started in the other command session) was in progress, then would resume when the local-PED operation terminated, or

- the ped connect command would begin executing, would pause while the local-PED operation was in progress, and would eventually time-out if the local-PED operation did not terminate sufficiently quickly.

If a Remote PED connection is currently in force, then the local PED is ignored, and all PED requests are routed to the Remote PED.

If a Remote PED connection is currently in force, then subsequent attempts to start a different connection are refused until the current connection times out or is deliberately stopped.

Remote PED Timeout

In local PED mode, one Luna PED is connected directly to the HSM. Timeouts are governed by the configuration of the host and HSM.

In Remote PED mode, the PED Server on each remote Workstation has a timeout setting (which can be modified), and the HSM has a Remote PED timeout setting that can be seen and modified in the config file. If nothing has been set, then the default value for the Remote PED connection timeout (1800 seconds) is in effect.

The Remote PED server instances on workstations, and the Remote PED client inside the Luna HSM host are not aware of each others' timeout values. For a given Remote PED connection, the shorter timeout value rules. Thus, if a Remote PED server on one of your workstation computers were to timeout during a Remote PED sequence, it would log the event and send a message to the HSM host that the connection had been open too long. The Remote PED Client on the Luna HSM host, receiving that message, would gracefully close the link and the appliance-side timeout would not be reached.

Ports

We suggest port 1503 for the Remote PED connection, but you can use any port that does not conflict with another operation.

Windows 7

PedServer.exe (on the computer to which your Remote PED is attached) is run from the command line.
To use PedServer on a Windows 7 computer, right-click the Command Prompt icon, and from the resulting menu select "Run as Administrator".

If you lack system permissions to operate as Administrator on the computer that is to host the PED Server, contact your IT department to address the situation.

If you open a command-prompt window as an ordinary user in Windows 7, and run PedServer.exe, the program detects that it lacks access and permissions, and returns an error like the following:

```
C:\Program Files\SafeNet\LunaClient>pedserver mode start
Ped Server Version 1.0.5 (10005)

Failed to load configuration file.  Using default settings.
```

```
Ped Server launched in startup mode.
Starting background process
InternalRead: 10 seconds timeout
Failed to recv query response command: RC_OPERATION_TIMED_OUT c0000303
Background process startup timed out after 10 seconds.
Startup failed. : 0xc0000303 RC_OPERATION_TIMED_OUT

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

If you encounter the error above, use Windows Task Manager to select the PedServer process, right-click, and select "End process", before cleanly retrying PedServer.exe via an Administrator Command Prompt.

Other Windows versions have not exhibited this requirement.

Limitations

The connection is one-on-one. While a Remote PED connection is active between one HSM and one remote PED workstation (running PedServer.exe), neither entity is able to make a similar connection with a different partner. The connection must time out, or be deliberately stopped before the HSM can connect with another PedServer workstation and enter a new remote PED authentication arrangement.

When an RPV is created, it is a randomly-generated value that exists nowhere else. You control which (and how many) HSMs will contain that RPV, and which (and how many) orange RPK PED Keys will contain copies of it. A Remote PED with an inserted RPK (orange Remote PED Key) can be used only with distant Luna HSMs that share that exact RPV. If you launch a Remote PedServer with a connected Remote PED and provide any other orange PED Key, it is not accepted by any distant Luna HSM that does not have the matching RPV. In this manner, you can segregate the ability of personnel to remotely control specific HSMs, by controlling which orange PED Keys they are issued. Two people in the same office could have access and control of entirely different sets of remotely located HSMs, with no overlap, as long as you trusted them not to exchange orange PED Keys. You can further control the extent of each holder's access by invoking MofN when you first create an RPV.

Compatibility

Remote PED for Luna HSM 5.2 is not compatible with earlier HSM versions.

Remote PED and pedclient and pedserver

When it is not convenient to be physically near the host computer that contains a Luna PCI-E HSM, in order to connect a Luna PED and present PED Keys, you can operate remotely and securely, as follows:

- On the host computer (which can run Windows, Linux, Solaris, HP-UX - see the current OS support table in the Customer Release Notes) containing the Luna PCI-E HSMs, allow remote desktop access or ssh, and have the `pedclient.exe` program available.
- On the remote administrative workstation (which for this purpose must run the Windows operating system) use remote-desktop client or use ssh, have a Luna PED2 (with Remote capability) connected, and have the SafeNet Luna `pedserver` tool installed and running.
- Make the Remote PED connection between the host and the remote administrative workstation. Start the pedserver listening on the workstation. Start pedclient on the host (containing the HSM)
- Make the remote desktop or ssh connection between the workstation and the host computer, and run `pedclient.exe` on the host computer, indicating the slot number of the HSM for which Remote PED services are to be provided. The combination of `pedserver` on one computer and `pedclient` on the other provides the

trusted path for secure transfer of authentication data.

- Run commands on the HSM (on the host computer) via the remote desktop or ssh

Use **static IP addressing** for PED Client / PED Server. PED Client can fail to find a server if a dynamic address is indicated.

An example error might look like this:

```
lunash:>hsm ped connect -ip 192.20.11.67 -port 1503
Luna PED operation required to connect to Remote PED - use orange PED Key(s).

Ped Client Version 1.0.5 (10005)
Ped Client launched in startup mode.
readIPFromConfigFile() : config file did not contain an IP address.
Startup failed. : 0xc0000404 RC_FILE_ERROR
Command Result : 65535 (Luna Shell execution)
lunash:>
```

Security of Remote PED

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 merely provide the communication pathway between the PED and the HSM. Along that path, the authentication data remains encrypted.

Multiple HSMs and Remote PED

A host computer with multiple PCIe slots (the slots must be x4 or larger and not dedicated for video card operation) can accept and operate multiple Luna PCI-E 5 HSMs.

Remote PED (via `pedclient.exe`) can communicate - can provide PED services - to one Luna PCI-E HSM in your host computer at any one time (pedclient sees each HSM as a numbered slot).

To provide PED interaction (remotely) to another Luna PCI-E HSM in that same host computer, you must close pedclient.exe (on your remote workstation) for that first slot/HSM and then open `pedclient.exe` for the next slot/HSM.

Once a Luna PCI-E HSM (a slot) has been set up with its authentication data cached (autoActivation), and `pedclient` has closed (perhaps because you need to open `pedclient` for another HSM in your host computer), you must not issue any command to that original slot that would require PED interaction.

If you issue a command that invokes a PED operation, when no PED is connected to the HSM (such as when `pedclient` and the Remote PED are busy with another HSM in your host computer, or when `pedclient.exe` is simply not running), the affected HSM pauses until the requested operation times out. This means that any client application that was using that HSM stops for the duration of the timeout.

Configuring Remote PED

Luna PED is a Luna accessory device that allows compatible Luna HSMs to securely store their authentication data on PED Keys (specially configured USB tokens), to retrieve that data when needed, and to modify the content of PED Keys for security and operational purposes. All of the Luna PED and PED Key actions can be accomplished with the

Luna PED directly connected to the Luna HSM, and powered by that HSM. Sometimes that direct connection is inconvenient, due to location of the HSM and of the personnel who are charged with controlling and managing the HSM. In such circumstances, it can be useful to employ a Luna PED with Remote capability.

Remote PED is supported (and requires installation/configuration) in two parts:

- PEDClient, which runs on the Luna HSM host computer and allows the HSM to seek PED Key data from a remotely located Luna PED, via
- PEDServer, which runs on a workstation, laptop, or server computer to which a Remote-capable Luna PED is USB connected.

PEDClient is part of the LunaClient software installation for every type of Luna HSM except Luna SA (because PEDClient is already present within the Luna SA appliance).

PEDServer is installed if the "Remote PED" option is selected during LunaClient software installation, and includes the PedServer.exe executable, along with the SafeNet Luna PED device drivers. If the target computer is intended to be a PEDServer, but is not going to be a Client to your Luna HSM, then you do not need any of the other LunaClient software; you can use LunaClient installer to install only the Remote PED option.

You will need:

- Your HSM host, configured as described elsewhere in this document, with PEDClient available, and with its own working network connection.
- A suitable laptop, workstation, or server computer, for PEDServer, with a supported operating system (see the Customer Release Notes for supported platforms)
- Sufficient privileges on the PEDServer computer, depending on platform and location (local network, WAN, VPN...)
- Current LunaClient installer (LunaClient.msi)
- Luna PED (Remote capable) V.2.5.0-2 or newer (see the bottom of the PED's Select Mode menu for the version)
- The power block and cord that accompanied your Remote PED, and the USB-A to USB-Mini-b cable
- PED Keys.
- A network connection.

Configure the PEDClient and PEDServer

This configuration takes place in two locations:

- on the computer that has the HSM,
- on the computer that is to have the Luna Remote PED.

HSM-Side

1. Install/configure your HSM host as described previously.
2. Change to the directory where LunaClient is installed and launch lunacm.

   Type: `c:\Program Files\SafeNet\LunaClient>` **`lunacm`**
3. With a Luna PED connected **locally**, initialize a Remote PED Vector for the HSM and for an orange PED Key.

By means of your responses to the PED prompts, you can choose to have the HSM generate a new RPV to be held by both the HSM and a new orange PED Key, or you can choose to re-use an RPV already on an existing orange PED Key, and imprint that on the HSM.

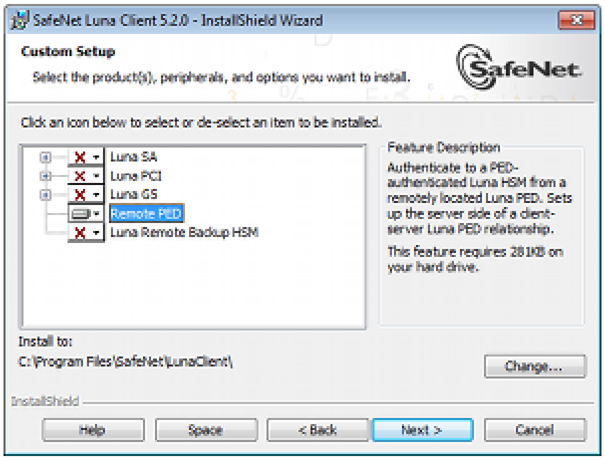
As always, we suggest that you make at least one extra copy of the Remote PED Key.

Type: `lunacm:>` **`ped vector init`** and respond to the Luna PED prompts.

4. Bring an orange PED Key, containing the RPV for this HSM, from the HSM to the location of the Remote PED server.

PEDServer-Side

1. Install LunaClient software, selecting "Remote PED" option - for the purposes of Remote PED, any additional LunaClient installation choices are optional for this host system.



SafeNet Luna Client 5.2.0 - InstallShield Wizard

Custom Setup

Select the product(s), peripherals, and options you want to install.

Click an icon below to select or de-select an item to be installed.

- Luna SA
- Luna PCI
- Luna G5
- Remote PED
- Luna Remote Backup HSM

Feature Description

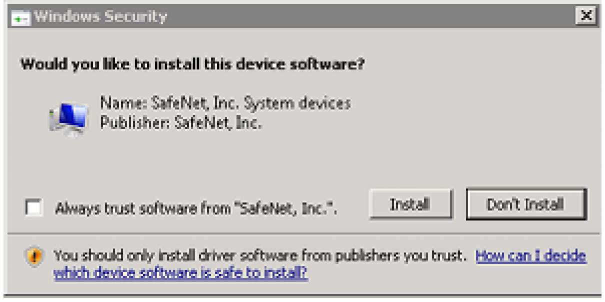
Authenticate to a PED-authenticated Luna HSM from a remotely located Luna PED. Sets up the server side of a client-server Luna PED relationship.

This feature requires 281KB on your hard drive.

Install to:

C:\Program Files\SafeNet\LunaClient\

2. Luna PED should not yet be connected to the PEDServer computer.



Windows Security

Would you like to install this device software?

Name: SafeNet, Inc. System devices
Publisher: SafeNet, Inc.

☐ Always trust software from "SafeNet, Inc.".

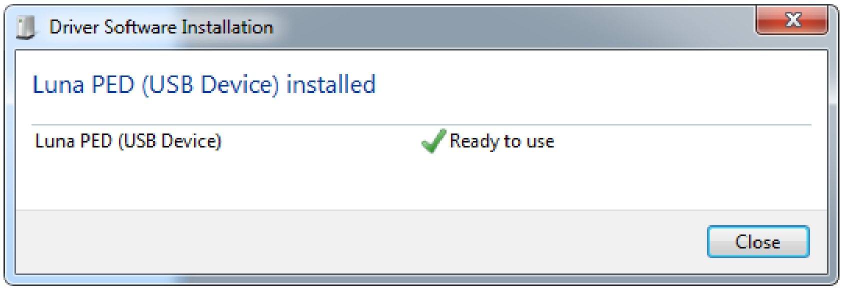
You should only install driver software from publishers you trust. How can I decide which device software is safe to install?

Select [Install] when prompted to install the driver.

3. Reboot the computer to ensure that the LunaPED driver is accepted by the operating system. This is not required for Windows Server Series.

4. Connect the Remote Capable Luna PED to AC power, using the supplied power block, and to the PEDServer computer, using the supplied USB-A to USB-mini-b cable.

Windows acknowledges the new device.



Luna PED (USB Device) installed

Luna PED (USB Device) ✓ Ready to use

5. Luna PED performs its start-up sequence, and settles into Local Mode, by default.

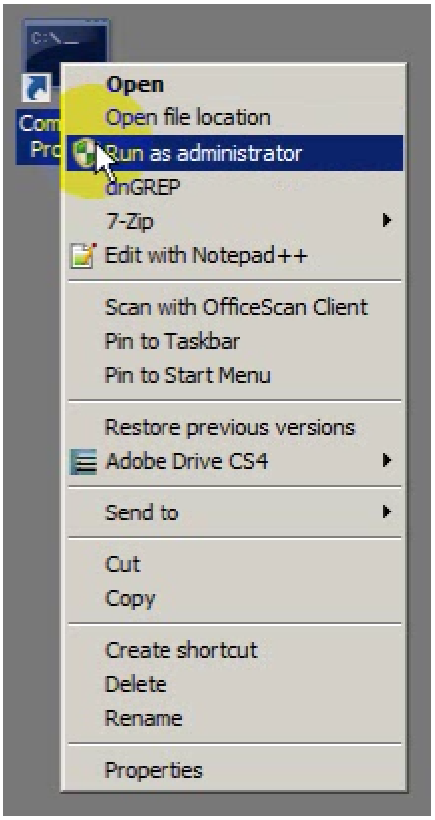
   Press the [<] key to access the "Select Mode" menu.

6. Press [7] to select "Remote PED" mode.

7. Ensure that your organization's Firewall does not block communication between pedclient and PEDServer. If switching off the firewall for Home and Public Network is not an option, see the Troubleshooting section below.

8. Open a Command Prompt window.

   If PedServer.exe attempts to access the pedServer.ini file in c:\Program Files\.... that is treated as an action in a restricted area in some versions of Windows. In that case, you should open the Command Prompt as Administrator, rather than as your normal user. To do so, right-click the Command Prompt icon and, from the pop-up menu, select **Run as administrator**.



> **Note:** Windows Server 2008 launches Command Prompt as Administrator, by default, so no special steps are necessary.

> **Note:** By default, PedServer.exe attempts to access pedServer.ini if such a file exists in the expected location. If it does not exist, then default values are used by PedServer.exe until you perform a "-mode config -set" operation to create a pedServer.ini.

9. Go to the installed LunaClient directory.

   Type `cd "\Program Files\SafeNet\LunaClient"`

10. Launch the PEDServer.

    Type `pedserver -mode start`

11. Verify that the service has started.

    Type `pedserver -mode show` and look for mention of the default port "1503" (or other, if you specified a different listening port).
    As well, "Ped2 Connection Status:" should say "Connected". This indicates that the Luna PED that you connected (above) was found by PEDServer.

> **Note:** If a port other than the default 1503 was specified in `pedserver -mode start`
> ... like `pedserver -mode start -port 1523`
> then `pedserver -mode show` command should pass in the same port
> ... like `pedserver -mode show -port 1523`.
>
> If a non-default value for the listening port was configured (meaning that it was present in pedServer.ini), then `pedserver -mode show` finds the port from that file.

12. Note the IP address of the PEDServer host. We generally recommend using static IP, but if you are operating over a VPN, you will likely need to ascertain the current address each time you [re-]connect to the VPN server and are assigned an address.

```
C:\windows\system32>ipconfig

Windows IP Configuration


Ethernet adapter Bluetooth Network Connection 2:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Wireless LAN adapter Wireless Network Connection 4:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Wireless LAN adapter Wireless Network Connection 3:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::cd74:173c:692a:22b0%26
   IPv4 Address. . . . . . . . . . . : 192.168.0.16
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.0.1

Ethernet adapter Local Area Connection 3:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Ethernet adapter Local Area Connection 2:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::5456:b034:a1ff:96fe%14
   IPv4 Address. . . . . . . . . . . : 182.16.153.114  <<--- this one in our example
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . :

Tunnel adapter isatap.{9EE24CB0-63D2-4D40-902B-3DC3193701FA}:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Tunnel adapter Local Area Connection* 17:

   Connection-specific DNS Suffix  . :
   IPv6 Address. . . . . . . . . . . : 2001:0:9d38:90d7:3cca:2f17:3f57:ffef
   Link-local IPv6 Address . . . . . : fe80::3cca:2f17:3f57:ffef%11
   Default Gateway . . . . . . . . . : ::

Tunnel adapter isatap.{9D552290-62C3-479B-A312-FAEA518B1655}:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Tunnel adapter isatap.{184652AE-5DF0-470C-84BE-B4D09760D3C9}:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

C:\windows\system32>
```

Note: Your organization's VPN might be configured with a relatively short lease time, so that you might need to re-establish the Luna Remote PED connection at intervals of hours or days, providing the newly assigned IP address of your PEDServer computer each time.

HSM-Side

Note: For the purposes of the PEDClient (the HSM that seeks a Remote PED connection) you can specify the PEDServer's IP address and listening port each time you connect. Or you can use the `lunacm:>` **`ped set`** command to configure either, or both of those parameters, which are then picked up by the `lunacm:>` **`ped connect`** command when you wish to establish the connection.

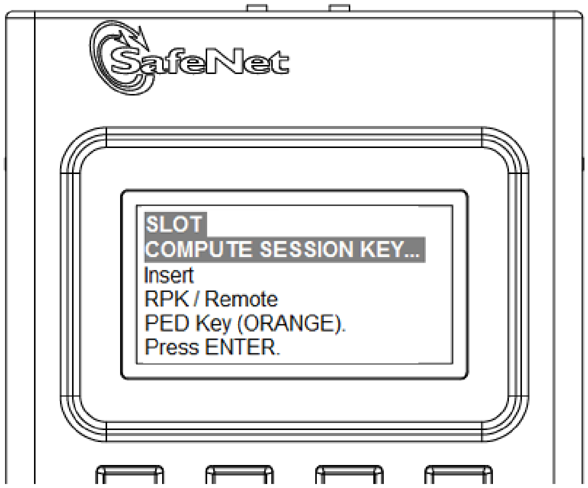
If the listening port of the PEDServer is not specified, then the default value "1503" is assumed. The IP address must be specified somewhere; there is no original default. If an IP address or a port is specified in the `lunacm:>` **`ped connect`** command, it overrides any value that was set by `lunacm:>` **`ped set`** , but only for the current connection.

1. Launch the PEDClient on your HSM server, identifying the PEDServer instance (configured above) to which the HSM is to connect for its authentication requirements.

   Type `lunacm:>` **`ped connect -ip`** `<pedserver ip>` **`-port`** `<pedserver listening port>`
   (substituting your actual PEDServer IP and port)
   for example: `lunacm:>` **`ped connect -ip 182.16.153.114 -port 1503`**

   Luna PED operation required to to connect to Remote PED - use orange PED Key(s).

   At this point, the remote Luna PED should come to life, briefly saying "Token found..." followed by this prompt:



```
SLOT
COMPUTE SESSION KEY...
Insert
RPK / Remote
PED Key (ORANGE).
Press ENTER.
```

2. Insert the orange PED Key that you brought from the HSM to the remote PED, and press [Enter] on the PED keypad.

   When the orange PED Key is accepted, control returns to the HSM command-line with a success message: "Command Result : 0 (Success)"

Once you have reached this point, you can continue to issue HSM or Partition commands, and whenever authentication is needed, the Remote PED will prompt for the required PED Key and associated key-presses.

Relinquishing Remote PED

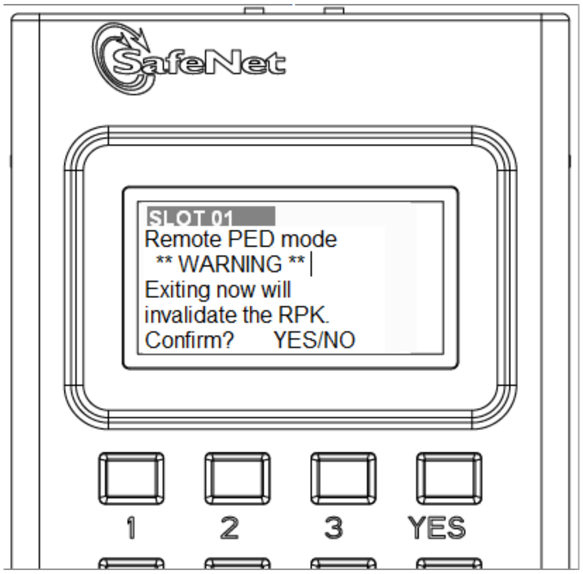
The PEDServer utility continues to run until explicitly stopped.

On the HSM end, PEDClient (launched by the "connect" command) continues to run until you explicitly stop with the "disconnect" command, or the link is broken. At any time, you can run the command in "show" mode to see what state it is in.

If you physically disconnect the Remote PED from its host, the link between PEDClient and PEDServer is dropped.

If the network connection is disrupted, or if your VPN closes, the link between PEDClient and PEDServer is dropped.

If you attempt to change menus on the Remote PED, the PED warns you:



```
SLOT 01
Remote PED mode
  ** WARNING **
Exiting now will
invalidate the RPK.
Confirm?     YES/NO
```

If you persist, the link between PEDClient and PEDServer is dropped.

If the "IdleConnectionTimeoutSeconds" is reached, the link between PEDClient and PEDServer is dropped. The default is 1800 seconds, or 30 minutes. You can modify the default value with the "-idletimeout" option.

Any time the link is dropped, as long as the network connection is intact (or is resumed), you can restart PEDClient and PEDServer to reestablish the Remote PED link. In a stable network situation, the link should remain available until timeout.

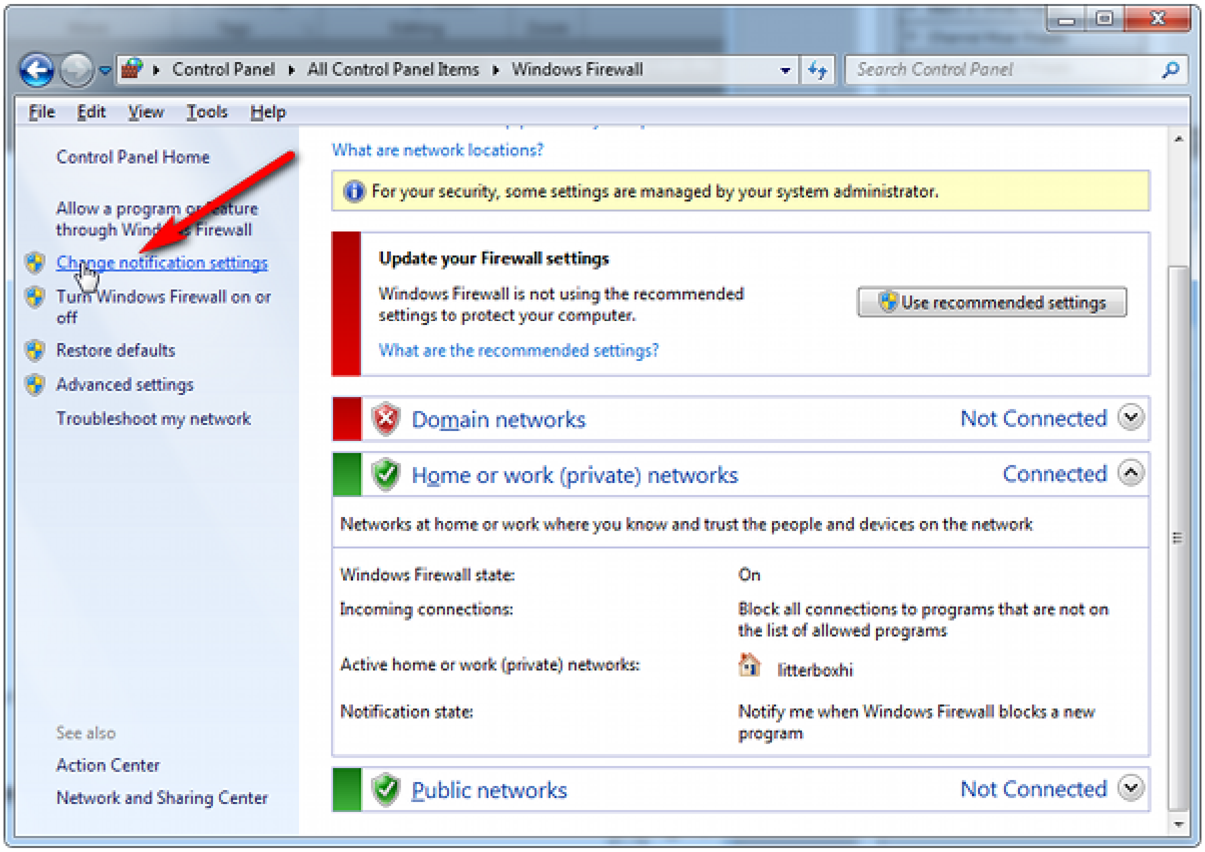
Troubleshooting

Here are some suggestions for addressing some possible issues when configuring Luna Remote PED.

Firewall blocking

If you experience problems while attempting to configure a Luna Remote PED session over VPN, you might need to adjust Windows Firewall settings.

1. From the Windows Start Menu, select "Control Panel".
2. From the "Control Panel", select "Windows Firewall".
3. From the "Windows Firewall" dialog, 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
What are network locations?
Use recommended settings
What are the recommended settings?

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.

Domain networks — Not Connected

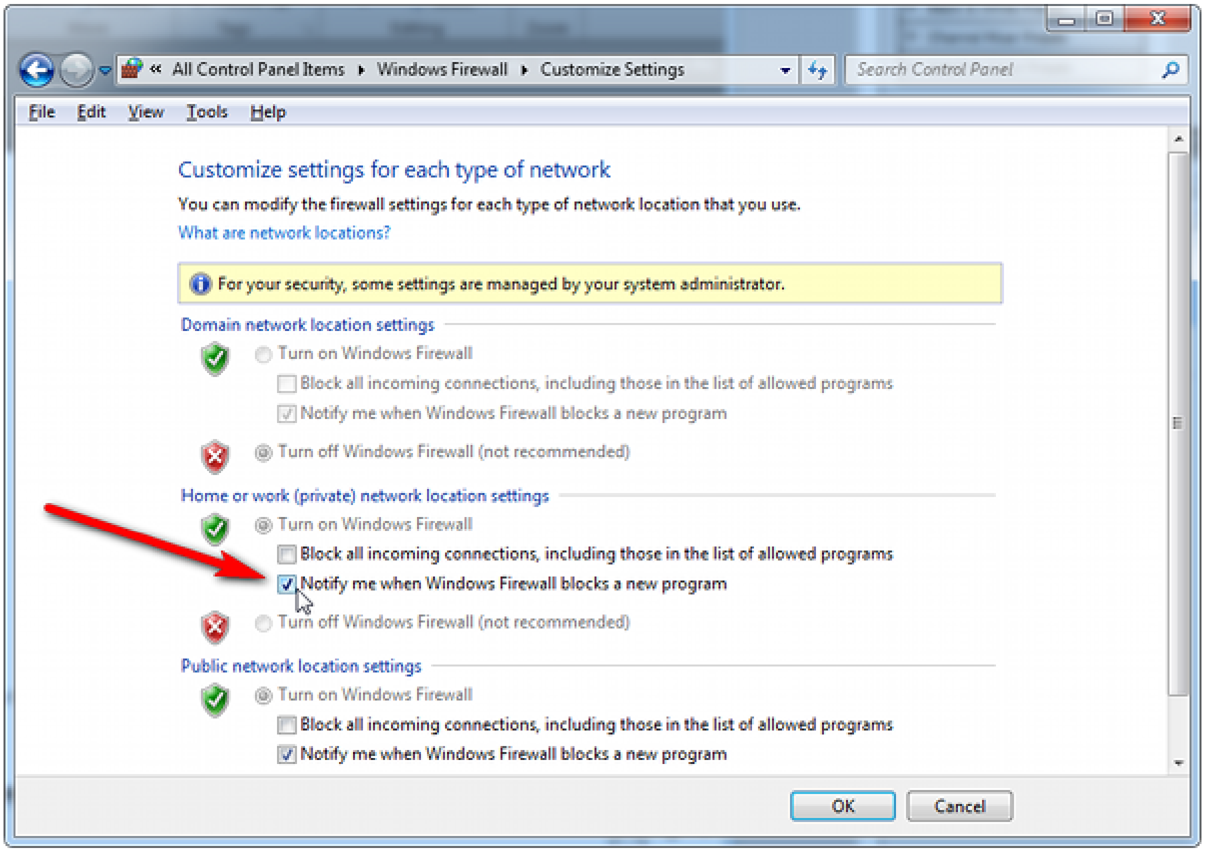
Home or work (private) networks — Connected

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

Without this setting, it might not matter that you have Administrator-level privileges on the PEDServer host computer, because Windows would silently block the connection from PEDClient to PEDServer, and not give you an opportunity to exercise your power to approve the connection.

With notification turned on, a dialog box pops up whenever Windows Firewall blocks a program, allowing you to override the block, which permits the Luna Remote PED connection to successfully listen for PEDClient connections.

Port Access

Another possible issue is that some networks might be configured to block access to certain ports. If such policy on your network includes ports 1503 (the default PEDServer listening port) and 1502 the administrative port, then you might need to choose a port other than the default, when starting PEDServer, and similarly, when you launch the connection from the HSM end and provide the IP and port where it should look for the PEDServer. Otherwise, perhaps your network administrator can assist.

Using the Remote PED Feature

To use Remote PED for the first time, you will need:

- a Luna PED 2.4.0 (or later) with Remote PED feature installed (the SafeNet label on the back has the words "REMOTE PED CAPABLE", top center, between the SafeNet logo and the FCC declaration)

- a power adapter for the Remote PED (when the PED is not connected to a Luna SA, via the PED port, it requires the separate power adapter to supply its power - the USB connection is insufficient for that purpose)

- a complete set of PED Keys, including an orange Remote PED key (either new/empty or already containing a Remote PED vector)

- local access to the Luna SA (for the first session only)

- HSM/appliance that supports the Remote PED feature (includes the Remote PED Client)

- a workstation/PC with the PEDserver.exe (Remote PED Server application) running, and with the appropriate PED driver already installed [The software and driver are provided on the Luna SA Client CD, but are not automatically installed as part of the main Windows installer.

i) Browse to the Windows directory on the Client software CD, which contains sub-directories for Windows "32" and "64", as well as a directory labeled "remotePed".

ii) Enter the "remotePed" directory and double-click one of RemotePed32.msi or RemotePed64.msi installers, as appropriate for your platform. This moves the pedserver.exe software and the driver files onto your computer.

iii) When you connect your Luna PED2 Remote to electrical mains power (AC power outlet) and to your computer's USB port, the operating system detects the new hardware and asks you to locate the appropriate driver. Use the dialog to browse to the location where the LunaPED driver has been placed by the installer.].

You will need physical access to your Luna SA when first setting up Remote PED, because the Remote PED vector must be created by the HSM and imprinted on a blank PED Key, or it must be acquired from a previously imprinted orange PED Key and stored in the HSM. Thereafter, the orange PED Key is used with the Remote PED from a remote location, and the connection is secured by having the matching Remote PED vector at both the HSM and the Remote PED server (your remote workstation with Remote PED attached).

If you encounter timeout problems (possible if you are using M of N with many keys, or if you are reading instructions as you go, or are otherwise not speedy while following prompts), you can adjust timeout values to allow for a more relaxed pace. For PedServer.exe, you can do:
pedserver -mode config set -socketreadrsptimeout <seconds>
but you would also need to increase the timeout in the crystoki.ini client software configuration file.
Moreover, the PEDServer -socketreadrsptimeout must always be larger than the timeout in the configuration file.

In general, do not change settings (especially in the crystoki.ini file) unless you have good reason to do so, or are instructed to do so, by SafeNet Customer Support.

Use **static IP addressing** for PED Client / PED Server. PED Client can fail to find a server if a dynamic address is indicated.

An example error might look like this:

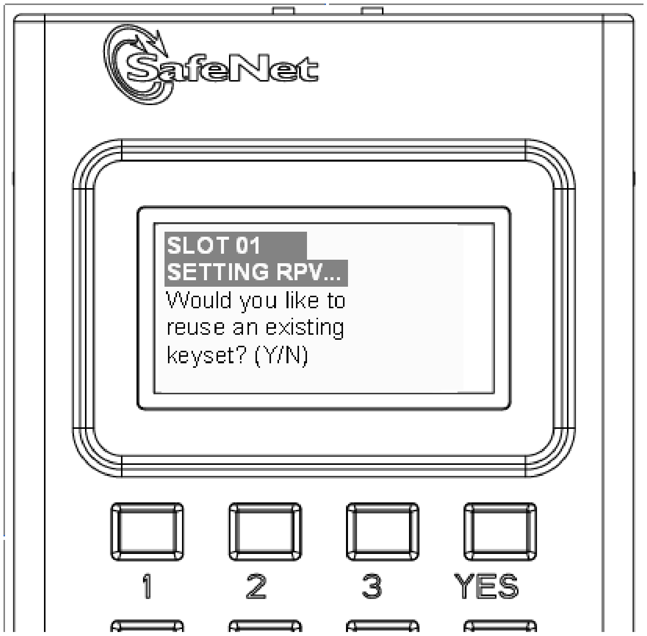
```
lunash:>hsm ped connect -ip 192.20.11.67 -port 1503
Luna PED operation required to connect to Remote PED - use orange PED Key(s).

Ped Client Version 1.0.5 (10005)
Ped Client launched in startup mode.
readIPFromConfigFile() : config file did not contain an IP address.
Startup failed. : 0xc0000404 RC_FILE_ERROR
Command Result : 65535 (Luna Shell execution)
lunash:>
```

Setup Instructions

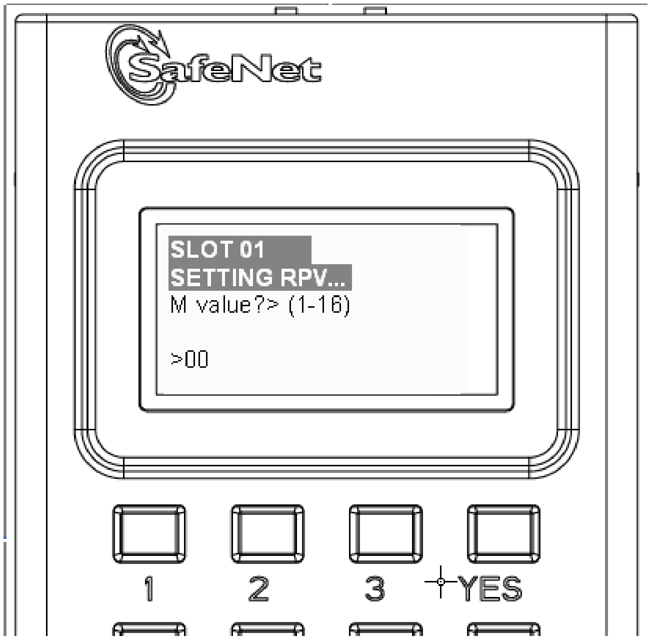
The steps to set up Remote PED are:

1. In network configuration for your Luna SA appliance, ensure that the second Ethernet port is configured for use. [In order to properly enable the Remote PED capability, the second Ethernet port must be configured, either to a real address, if you intend to use it, or to a dummy address. Here is an example:
   ```
   lunash:>network interface -ip 192.168.1.254 -netmask 255.255.255.0 -dev eth1
   ```
   You will need to restart NTLS to have the change take effect.
   ```
   lunash:>service restart ntls ]
   ```
2. Initialize the HSM [if you have not already done so]- the creation of the orange Remote PED key requires HSM login; HSM login requires an initialized HSM, all of which must be done with a local PED connection the first time.
3. Have the Luna PED connected to the PED port of the HSM, and set to Local PED mode.
4. Login as SO:
   ```
   [myluna] lunash:>hsm login
   Luna PED operation required to login as HSM Administrator - use blue PED key(s).
   'hsm login' successful.
   Command Result : 0 (Success)
   [myluna] lunash:>
   ```
5. Have a blank PED Key, with orange label, ready. Create and imprint the RPV (Remote PED Vector):
   ```
   [myluna] lunash:>hsm ped vector init
   WARNING !! This command will initialize remote PED vector (RPV).
   If you are sure that you wish to proceed, then enter 'proceed', otherwise this
   command will abort. > proceed
   Proceeding...
   Luna PED operation required to initialize remote PED key vector - use orange PED
   key(s).
   (At this time, go to the Luna PED and respond to the prompts by providing either
   a "fresh" orange PED key (which prompts creation and imprinting of a new/unique
   RPV) or an already-imprinted orange PED Key (which prompts the PED to ask you to
   reuse the existing PED Key data), along with additional blanks if you intend to
   make duplicates.)
   ```
   The PED says:



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

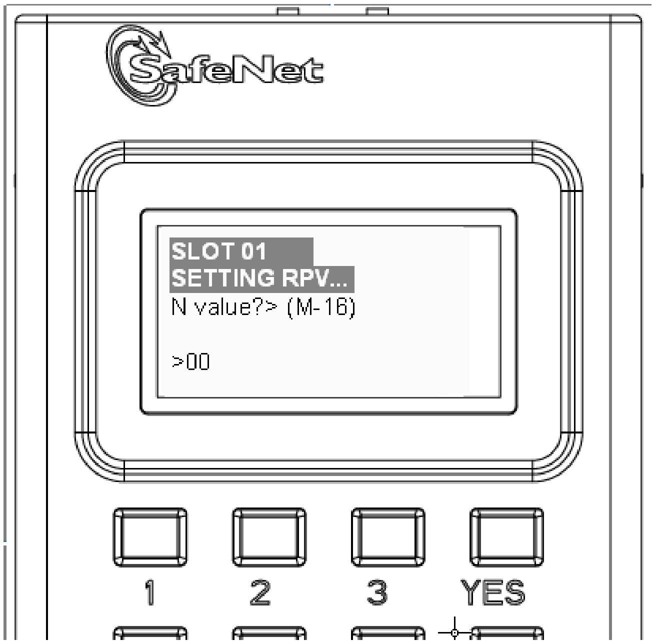
If this is the first RPV that you are creating, then answer [NO].
If you have an existing RPV on an orange PED Key, then answer [YES] if you want to preserve it and add it to this current HSM, or [No] if you have made a mistake and wish to find a different blank (or outdated) key to imprint.
For this example, we will assume no existing RPV.
The PED says:



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

>00
```

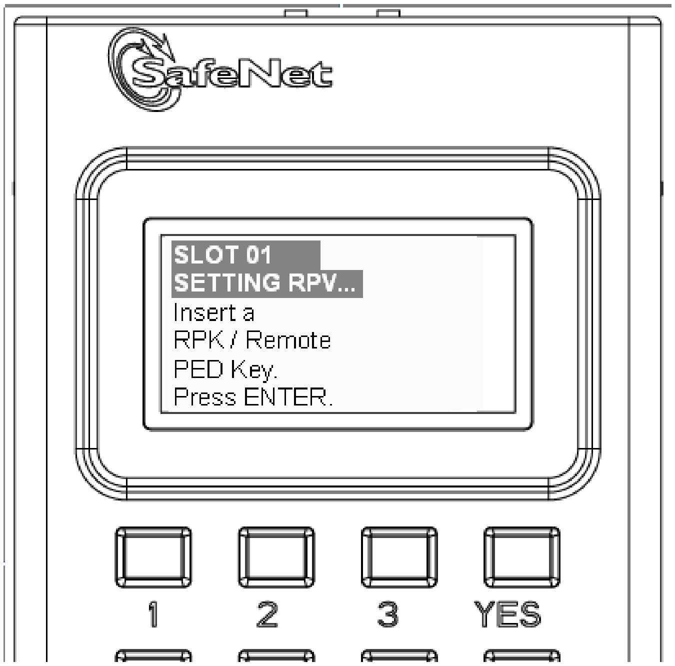
If you wish to split the RPV secret over several RPKs, for MofN split-knowledge, multi-person access control of the Remote PED function, then input a value for M that is greater than "1". This is the number of persons - each holding an orange key containing a split of the RPV secret - who must come together and present their portions whenever the RPK is required. If you prefer not to invoke MofN, then press [1], followed by [Enter].



SLOT 01
SETTING RPV...
N value?> (M-16)

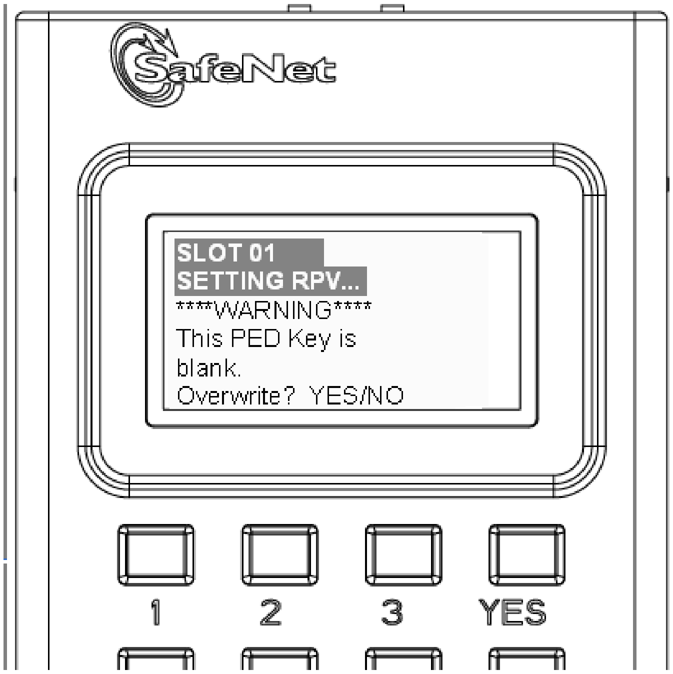
>00

If you have invoked MofN with an M value greater than "1", then you must enter a value for N that is equal to, or greater than, M. N is the total pool of orange keys over which your RPV will be split, from which sub-groups of quantity M will be required for authentication. The simplest scheme is to declare a value for M that gives you the desired security oversight of the Remote PED function, and then specify N slightly larger so that you can always have at least quantity M key-holders available, even when some are absent for vacation, travel, illness or other reasons. Example: M=3, N=5, where any 3 of the total 5 splits can combine to reconstitute the secret.



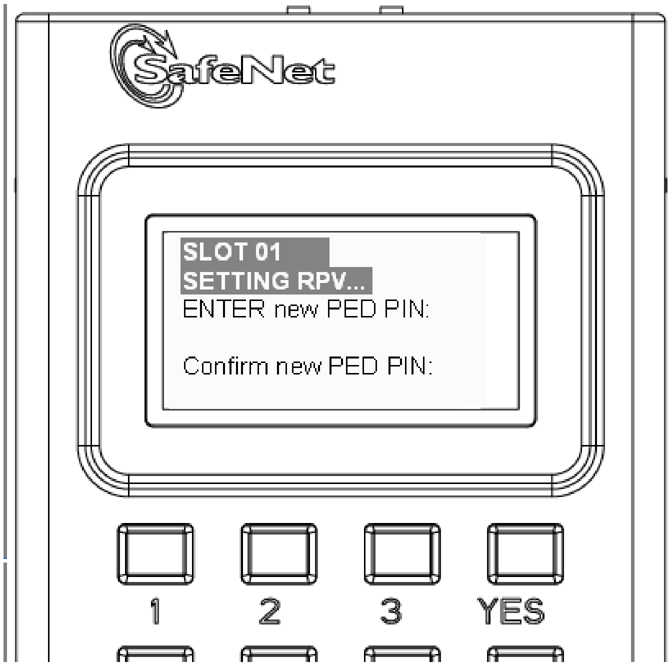
SLOT 01
SETTING RPV...
Insert a
RPK / Remote
PED Key.
Press ENTER.

Do as prompted, inserting an unused PED Key into the PED's key slot (top-right of the PED), and press [ENTER].
For a fresh, new, never-before imprinted PED Key, the PED says:



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

Answer [YES] so that the HSM can create an RPV and transfer it to the PED, where it is imprinted onto the blank PED Key that you have inserted. If you invoked M of N, then the PED will prompt you to continue inserting orange PED Keys for imprinting with portions of the secret until you have imprinted quantity N of them.

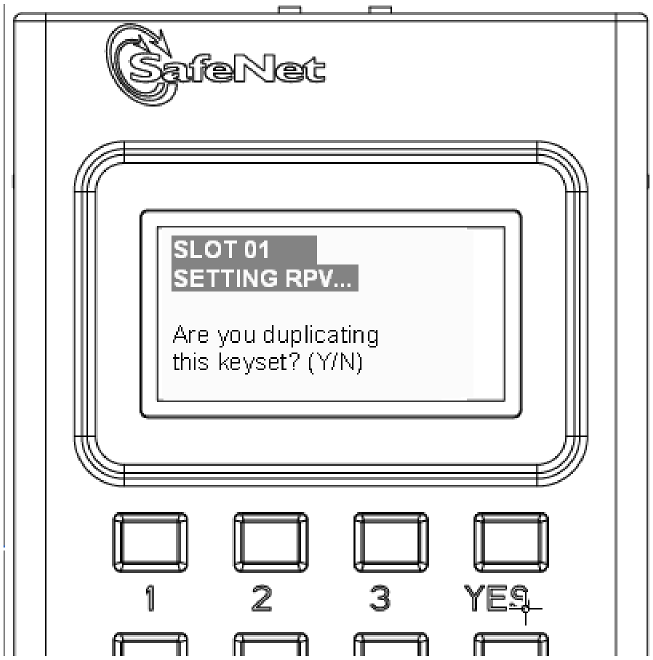


If you need two-part security to protect the Remote PED function, and wish to add a "something you know" component to the "something you have" (physical PED Key), type a series of digits on the keypad, then type them again to confirm. Now, whenever you are prompted to present the orange RPK, you must also input the code - called a PED PIN - that you have just added. The secret that unlocks the HSM to perform Remote PED operation is now a combination of a data secret contained in the physical key, and a typed-in numeric code that you must remember.

Press [Enter] with no digits, if you do not wish an additional "something you know" secret attached to this PED Key. In future, Luna PED will nevertheless prompt you for a PED PIN whenever you present the RPK, but you will always just press [Enter] (with no digits) at that prompt - no PED PIN required.

This completes the imprinting of the key (or keys if you opted for MofN).

While the imprinted orange PED Key is still in the PED's slot, Luna PED then wants to know if you intend to make some copies of the currently-inserted PED Key (that now carries the RPV for the HSM) or group of PED Keys:



Answer [YES] if you wish to make copies, and follow the instructions to insert keys and press ENTER. Respond to the prompts about overwriting, and PED PIN, etc.
When you have made all the copies that you wish, respond [NO] to the final prompt.
Control is returned to the lunash command line.

```
Ped Client Version 1.0.0 (10000)
Ped Client launched in shutdown mode.
Ped Client is not currently running.
Shutdown passed.
Command Result : 0 (Success)
[myluna] lunash:>
```

(If you see references to "shutdown mode", that's the Luna shell [lunash] exchanging messages with the Remote PED Client application (also found on your Luna appliance), which is called, runs in the background, and shuts down, possibly multiple times, depending upon the task that you have initiated via [lunash:>] commands.)

6. At this point, you have an HSM with an RPV (Remote PED Vector) set, and one or more orange PED Keys carrying that same RPV. Bring a SafeNet Luna PED 2 with Remote PED capability, the PED Keys (blue and black and red), and at least one imprinted orange PED Key to the location of your workstation computer (anywhere in the world with a suitable network connection). You should already have the most recent PED driver software and the PedServer.exe software installed on that computer

   [The software and driver are provided on the Luna SA Client CD, but are optional during the installation process. If you intend to use Remote PED (and therefore need the PED driver and the PedServer executable program, ensure that Remote PED is among the options selected during installation. Alternatively, you can launch the installer at a later time and modify the existing LunaClient installation to include Remote PED at that time.
   When you connect your Luna PED2 Remote to electrical mains power (AC power outlet) and to your computer's USB port, the operating system detects the new hardware and should locate the appropriate driver. If that does not

happen, then the system presents a dialog for you to help if find the location where the LunaPED driver has been placed.].

7. Connect the Remote PED to its power source via the power adapter.
8. Connect the Remote PED to the workstation computer via the USB cable.
9. When the PED powers on and completes its self-test, it is in Local PED mode by default.
   Press the [<] key to reach the "Select Mode" menu.
   Press [7] to enter Remote PED mode.
10. Open a Command Prompt window on the computer (for Windows 7, this must be an Administrator Command Prompt), locate and run PedServer.exe (we suggest that you try it out beforehand, to become familiar with the modes and options - if you experience any problem with PED operation timeout being too short, use "PedServer -mode config -set <value in seconds>" to increment the "sreadrsptimeout" value).
    Set PedServer.exe to its "listening" mode.

```
c: > PedServer -m start
Ped Server Version 1.0.5 (10005)
 Ped Server launched in startup mode.
Starting background process
Background process started
Ped Server Process created, exiting this process.
c:\PED\ >
```

NOTE: if you encounter a message "Failed to load configuration file...", this is not an error. It just means that you have not changed the default configuration, so no file has been created. The server default values are used.

11. Open an ssh session to the Luna SA appliance and login as admin.
12. Start the PED Client (the Remote PED enabling process on the appliance):

```
lush:> hsm ped connect -i 183.21.12.161 -port 1503
Luna PED operation required to connect to Remote PED - use orange PED key(s).
Ped Client Version 1.0.0 (10000)
 Ped Client launched in startup mode.
Starting background process
Background process started
 Ped Client Process created, exiting this process.
Command Result : 0 (Success)
 [luna27] lush:>
```

13. To verify that the Remote PED connection is functional, try some HSM commands that require PED action and PED Key authentication - the simplest is hsm login. First logout, because you were already logged in to the HSM...

```
 [luna27] lush:>hsm logout
 'hsm logout' successful.
Command Result : 0 (Success)
[luna27] lush:>hsm login
 Luna PED operation required to login as HSM Administrator - use Security Officer
(blue) PED key.
 'hsm login' successful.
Command Result : 0 (Success)
 [luna27] lush:>
```

14. At this point, you have successfully set up a Remote PED link between a workstation computer (with PED attached to its USB port) and a distant Luna SA/appliance. You have demonstrated that success by performing an HSM operation that demanded SO/HSM Admin PED Key authentication, without being physically near to the Luna

SA/appliance, and without having a Luna SA PED directly attached to the Luna SA/appliance.
You can now perform any HSM administration chores (including Cluster creation/administration) as though you were physically adjacent to the HSM, with equal confidence in the security of the system[1].

15. To disconnect:

```
[luna27] lush:>hsm ped disconnect
WARNING !! This command will disconnect remote PED.
If you are sure that you wish to proceed, then enter 'proceed', otherwise this
command will abort.
> proceed
Proceeding...
Ped Client Version 1.0.0 (10000)
Ped Client launched in shutdown mode.
Shutdown passed.
Command Result : 0 (Success)
[luna27] lush:>
```

Usage Notes

If a Remote PED session is in effect and you press the [<] key on the PED (to go to the PED's "Select mode" menu), that action amounts to exiting the Remote PED mode. Therefore, the PED displays a message:

```
** WARNING **
Exiting now will
invalidate the RPK.
Confirm ?    YES/NO
```

If you press [YES], the RPK-validated Remote PED session is dropped and must be re-established from the HSM (with "hsm ped connect <network-target>" before you can resume activity with the Remote PED.
In other words, if you want to use that PED for any other purpose than the current connection with one remote HSM, you have to drop the current session to make such other use of the PED, and then have the appropriate RPK available when you are ready to re-establish the prior Remote PED connection.)

The above note talks about a "session" that exists only between the Remote PED and the computer (actually the PedServer software running on that computer) to which the Remote PED is connected. That is separate from the session that was established between the distant appliance/HSM and the PedServer on your computer. The session between computer and HSM is time-sensitive - it is in existence while needed and is either dropped intentionally or times out after brief inactivity. The session between the Remote PED and its attached computer persists until you disconnect the PED or change modes, or until you stop the PedServer.exe process on the computer.)

PED KEY MIGRATION from older classic-PED Datakeys (the PED Keys that look like toy plastic keys) is NOT SUPPORTED over Remote PED, because the old classic PED 1.x has no way to connect to the PED Server. Migration of PED Keys from DataKeys to iKeys must be done locally.)

Here is an example of what you would see if the second Ethernet port is not configured

```
[mylunasa1] lunash:>hsm ped connect -i 172.20.10.135 -port 1503
Luna PED operation required to connect to Remote PED - use orange PED key(s).
Ped Client Version 1.0.5 (10005)
Ped Client launched in startup mode.
readIPFromConfigFile() : config file did not contain an IP address.
Starting background process
Background process failed to start : 0xc0000303 RC_OPERATION_TIMED_OUT
Startup failed. : 0xc0000303 RC_OPERATION_TIMED_OUT
```

[1] [HSM products that include Remote PED are now routinely submitted to approving agencies (like NIST/FIPS) for validation]

```
The remote PED connection is in a bad state. Please try again later.
Command Result : 65535 (Luna Shell execution)
[mylunasa1] lunash:>   )
```

If you encounter problems with Remote PED, "Troubleshooting Remote PED" on page 138.

Troubleshooting Remote PED

On a system with two network connections, if pedserver attempts to use an IP address that is not accessible externally, then command lunacm:>ped connect can fail.

Here is an example:

This host computer is accessible through 192.20.10.175 and has an additional IP address 192.168.72.1 (that is not accessible).

```
Ethernet adapter VMware Network Adapter VMnet8:
Connection-specific DNS Suffix . :
IP Address. . . . . . . . . . . . : 192.168.20.1
Subnet Mask . . . . . . . . . . . : 255.255.255.0
Default Gateway . . . . . . . . . :

Ethernet adapter Local Area Connection:

Connection-specific DNS Suffix . :
IP Address. . . . . . . . . . . . : 172.20.10.175
Subnet Mask . . . . . . . . . . . : 255.255.255.0
Default Gateway . . . . . . . . . : 172.20.10.10

Ethernet adapter Wireless Network Connection:

Media State . . . . . . . . . . . : Media disconnected

Ethernet adapter VMware Network Adapter VMnet1:

Connection-specific DNS Suffix . :
 IP Address. . . . . . . . . . . . : 192.168.72.1
Subnet Mask . . . . . . . . . . . : 255.255.255.0
Default Gateway . . . . . . . . . :
```

Command lunacm:> pedserver -m show returns

```
Ped Server Version 1.0.5 (10005)
Ped Server launched in status mode.

Server Information:
Hostname: noi1-502192
IP: 192.168.72.1
 Firmware Version: 0.0.0-0
PedII Protocol Version: 0.0.0-0
Software Version: 1.0.5 (10005)

Ped2 Connection Status: Disconnected
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: 5 (secs)
Server Idle Time: 5 (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%)
```

What to do

1. Ensure that Pedserver is listening on the IP address that is accessible from outside.
2. If that condition (step 1) is not the case then disable the network connection on which Pedserver is listening.
3. Restart Pedserver and confirm that Pedserver is listening on the IP address that is accessible from outside.

CHAPTER 12

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" on page 140
- "The SafeNet Chrysalis-UTSP MIB" on page 142
- "The SAFENET HSM MIB" on page 143
- "Frequently Asked Questions" on page 150

Overview and Installation

For Luna HSM 5.x, SafeNet supports Simple Network Management Protocol (SNMP v3) for remote monitoring of conditions on a local HSM that might require administrative attention.

MIB

We provide the following MIBs (management information base):

MIB Name	Description
CHRYSALIS-UTSP-MIB.mib	defines SNMP access to information about the Luna appliance
SAFENET-HSM-MIB.mib	defines SNMP access to information about the Luna HSM
SAFENET-GLOBAL-MIB.mib	must be found in your system path so that symbols can be resolved
SAFENET-APPLIANCE-MIB.mib	reports Software Version of Luna SA appliance (this MIB exists inside the appliance, only)

Copy all MIBs in <luna client install dir > to the MIB directory on your system.

Luna 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 Luna G5 and Luna PCI-E - for Luna SA, the subagent is already on the appliance.

The Luna SNMP subagent (luna-snmp) is an AgentX SNMP module that extends an existing SNMP agent with support for SafeNet HSM monitoring. It is an optional component of the Luna client installation. The subagent has been tested against net-snmp, but should work with any SNMP agent that supports the AgentX protocol.

SNMP subagent installation

After selecting one or more products from the main LunaClient installation menu, you are presented with a list of optional components, including the Luna SNMP subagent. It is not selected by default, but can be installed with any product except the Luna SA 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 Luna products that you wish to install, and include SNMP among your selections. The subagent is installed for any Luna product except Luna SA in isolation.
4. Proceed to Post-installation configuration.

Post-installation configuration

After the Luna 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 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 (ex. “service luna-snmp start” on Linux, or start SafeNet Luna 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.

luna-snmp.conf Options

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 manual 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)	15

Option	Description	Default
	disconnected.	
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

(The Chrysalis MIB is the SafeNet MIB for Luna 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)
hsmNonCriticalEvents	The total number of NON-critical HSM events that have been detected (any that are not among the critical list, above).
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.

Item	Description
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 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 3 172.20.11.59
SAFENET-HSM-MIB::hsmTable
```

Note: 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.

The information is defined in tables, as follows:

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

Item	Type	Description	Values
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
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
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
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

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 Luna 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 showp" and "hsm di" includes the following :

- SW Version
- FW Version
- HSM label
- Serial #
- HW Model
- Authentication Method
- RPV state
- FIPS mode
- HSM storage space (bytes)
- HSM storage space used (bytes)
- HSM storage free space (bytes)
- Performance level
- Max # of partitions
- # of partitions created
- # of free partitions
- Configuration (Cloning/CKE)
- License information similar to the output of the "hsm displayLicenses" command
- Policies as shown below.

```
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 special cloning certificate Disallowed
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 Allowed
FIPS evaluated Disallowed
Manufacturing Token Disallowed
Enable Remote Authentication Allowed
Enable forcing user PIN change Allowed
Enable portable masking key Allowed
Enable partition groups Disallowed
Enable remote PED usage Disallowed
Enable External Storage of MTK Split Disallowed
HSM non-volatile storage space 2097152
Enable HA mode CGX Disallowed
Enable Acceleration Allowed
Enable unmasking Allowed
Enable FW5 compatibility mode Disallowed
Unsupported Disallowed
Unsupported Disallowed
Enable ECIES support Disallowed
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 by changed by the HSM Administrator.
Changing policies marked "destructive" will zeroize (erase
completely) the entire HSM.
Description                       Value Code Destructive
===========                       ===== ==== ===========
Allow cloning                     On     7      Yes
Allow non-FIPS algorithms         On     12     Yes
SO can reset partition PIN        On     15     Yes
Allow network replication         On     16     No
Allow Remote Authentication       On     20     Yes
Force user PIN change after set/reset  Off    21     No
Allow offboard storage            On     22     Yes
Allow Acceleration                On     29     Yes
Allow unmasking                   On     30     Yes
```

Partition Information

At the HSM Partition level the information in the outputs of"partition show" and "partition showp" includes the following :

- Partition Name
- Partition Serial #
- Activation State
- AutoActivation State
- Partition storage space (bytes)
- Partition storage space used (bytes)
- Partition storage free space (bytes)
- Partition Object Count
- Partition Policies from the Partition showpolicies command

```
lunash:> partition showPolicies -partition mypartition
Partition Name: mypartition
Partition Num: 65038002

   The following capabilities describe this partition and can
   never be changed.

   Description                         Value
   ===========                         =====
   Enable private key cloning          Allowed
   Enable private key wrapping         Disallowed
   Enable private key unwrapping       Allowed
   Enable private key masking          Disallowed
   Enable secret key cloning                Allowed
   Enable secret key wrapping               Allowed
   Enable secret key unwrapping             Allowed
   Enable secret key masking                Disallowed
   Enable multipurpose keys                 Allowed
   Enable changing key attributes           Allowed
   Enable PED use without challenge         Allowed
   Allow failed challenge responses         Allowed
   Enable operation without RSA blinding    Allowed
   Enable signing with non-local keys       Allowed
   Enable raw RSA operations                Allowed
   Max failed user logins allowed           10
   Enable high availability recovery        Allowed
   Enable activation                        Allowed
   Enable auto-activation                   Allowed
   Minimum pin length (inverted: 255 - min) 248
   Maximum pin length                       255
   Enable Key Management Functions          Allowed
   Enable RSA signing without confirmation  Allowed
   Enable Remote Authentication             Allowed
   Enable private key unmasking             Allowed
   Enable secret key unmasking              Allowed
   Enable RSA PKCS mechanism                Allowed
   Enable CBC-PAD (un)wrap keys of any size Allowed


   The following policies are set due to current configuration
   of this partition and may not be altered directly by the
   user.

   Description                              Value
   ===========                              =====
   Challenge for authentication not needed  False


   The following policies describe the current configuration
   of this partition and may be changed by the HSM Administrator.

   Description                              Value        Code
   ===========                              =====        ====
   Allow private key cloning                On           0
   Allow private key unwrapping             On           2
   Allow secret key cloning                 On           4
   Allow secret key wrapping                On           5
   Allow secret key unwrapping              On           6
   Allow multipurpose keys                  On           10
   Allow changing key attributes            On           11
   Ignore failed challenge responses        On           15
   Operate without RSA blinding             On           16
   Allow signing with non-local keys        On           17
   Allow raw RSA operations                 On           18
   Max failed user logins allowed           10           20
   Allow high availability recovery         On           21
   Allow activation                         Off          22
   Allow auto-activation                    Off          23
   Minimum pin length (inverted: 255 - min) 248          25
   Maximum pin length                       255          26
```

```
   Allow Key Management Functions           On           28
   Perform RSA signing without confirmation On           29
   Allow Remote Authentication              On           30
   Allow private key unmasking              On           31
   Allow secret key unmasking               On           32
   Allow RSA PKCS mechanism                 On           33
   Allow CBC-PAD (un)wrap keys of any size  On           34


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

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 13

User and Password Administration

This chapter describes tasks related to identities in the HSM, including changing and resetting passwords. It contains the following sections:

About Changing Passwords

HSM

Resetting HSM (SO) Password

There is no provision to reset the Security Officer (SO) 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 any Partitions on that HSM.

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 you are an unauthorized person attacking the HSM without access to the password or PED Key), then the HSM is effectively compromised and must be re-initialized. To look at that another way, a user or SO who already has current authentication and just wishes to change that authentication, at his/her own level, is required to log in first (which protects against malicious changes), but resetting back to some default secret requires intervention by a higher authority. At the HSM level, there is no higher authority than the Security Officer / HSM Admin, so simple re-setting is not permitted.

If you re-initialize with the same cloning domain, you can, of course, restore from backup.

Changing HSM Password

To change the HSM password (for Password Authentication) or the secret on the SO PED Key (for Trusted Path), you must log in as SO using the current password (or SO PED Key).

Password Authenticated

```
lunacm:> hsm login -password <password>

Command Result : No Error

lunacm:> hsm changePw prompt -newpw <new_password> -oldpw <old_password>

Command Result : No Error
```

lunacm:>

The task is complete.

You may not set the Password to be "PASSWORD", which is reserved as the partition creation-time default only, and is too easy to guess for a real, operational, in-service password.

Trusted Path / PED Authenticated

If you issued the same command for a Trusted Path / PED Authenticated HSM, `lunacm` returns an error like "0x30 (CKR_DEVICE_ERROR)". The text passwords are not expected or wanted for this type of HSM.

For a Trusted Path / PED Authenticated HSM, do not include any text passwords in the command.

lunacm:> hsm changePw

Please attend to the PED

Luna PED prompts for the current blue SO PED Key.
After you insert that, and press [ENTER], Luna PED prompts for a new blue PED Key - that can be an entirely new iKey PED Key, or it could be the same one that you just used, now to be overwritten. If the key that you provide is blank, a new Security Officer secret is generated and imprinted on both the iKey PED Key and the HSM. If the key you provide has a valid ID on it, Luna PED says so, and asks if you wish to retain it or overwrite it. Once that is done, you are asked if you wish to make any additional copies of the new blue PED Key, and the task is finished. The HSM and Partition contents are intact, but anybody (and any application) that has only the old blue SO PED Key (or a copy of it) can no longer access the HSM for administrative actions.

During the PED interaction, you could elect to change the MofN status of the SO secret. That is, if (for example) you had not invoked MofN for your old blue-key secret, you could now set "N" to some number higher than 1, and "M" as well, which would have the effect of splitting your SO secret across "N" different blue PED Keys. In legacy Luna HSMs, this was not possible.

Partition

A deliberate change to a Partition password is different from a password reset(the command partition -resetpw -password <password> allows the SO to force a password change for the Partition -- this would be needed if the User had forgotten the Partition Password or if someone had made 10 bad login attempts; it would also be used in the case of personnel change. Note that an SO-settable policy determines whether the User can resume using the Partition with the new password, or the User is immediately forced to set his/her own new password before being allowed to resume using the Partition.) .

In both cases, the Partition or HSM contents remain intact.

Resetting Partition Password

- you must be logged in as SO, but
- you do not need to know the existing Partition password (for Password Authenticated systems) nor do you need to have the existing Partition Owner (black) PED Key (for Trusted Path Authenticated systems).

lunacm:> partition resetPw -password <new_password>

Changing Partition Password

- you do not have to be logged in as HSM Admin, but

- you do need to know the current password (for Password Authenticated) or have the current black User PED Key (for PED Authenticated HSM).

Password Authenticated

lunacm:> partition changePw -newpw <new_password> -oldpw <old_password>

The above works for a Password Authenticated HSM, and the task is finished. The Partition contents are intact, but anybody (and any application) that knows only the old password can no longer access the partition. For a Password Authenticated HSM the Partition Owner/User Password is also the Client password - your Client applications must be given the new password before they can resume using the Partition.

Trusted Path / PED Authenticated

If you issued the same command
(`lunacm:> partition changePw -newpw <new_password> -oldpw <old_password>`)
for a Trusted Path / PED Authenticated HSM, `lunacm` assumes that you wish to change the Partition challenge secret (if you previously created one). This is your opportunity to impose a new secret – of your own choosing – to replace the 16-character secret created for you by Luna PED. You might do this for convenience, or because your organization's security policy mandates regular password changes.

If you prefer not to expose the password in the clear, on-screen, you can issue the command as `lunacm:> partition changePw -prompt` which causes lunacm to prompt you for the old and new passwords, and hides your input with asterisks (*****...).

For a Trusted Path / PED Authenticated HSM, if you do **not** include the text passwords in the changepw command, then lunacm assumes that you wish to change the secret on the black PED Key.

lunacm:> partition changePw

Please attend to the PED

Luna PED prompts for the current black Partition User / Owner PED Key.
After you insert that, and press [ENTER], Luna PED prompts for a new black PED Key - that can be an entirely new iKey PED Key, or it could be the same one that you just used, now to be overwritten. If the key that you provide is blank, a new Owner/User secret is generated and imprinted on both the iKey PED Key and the HSM Partition. If the key you provide has a valid ID on it, Luna PED says so, and asks if you wish to retain it or overwrite it.. Once that is done, you are asked if you wish to make any additional copies of the new black PED Key, and the task is finished. The Partition contents are intact, but anybody (and any application) that has only the old black User/Owner PED Key (or a copy of it) can no longer access the partition for administrative or cryptographic activities.

Note that, on a PED Authenticated HSM, the Client challenge secret (that your Client applications present in order to access the Partition) is different and totally separate from the black Owner/User PED Key secret that permits administrative access to the Partition. By changing or re-setting the Partition Owner/User secret in the second example, you did not touch the Client Partition secret, which can still be used by Clients.
Similarly, you could use the command `lunacm:> partition createChallenge` to create a new Client secret (which must then be given to any Client application that needs to use the Partition), without affecting the black Owner/User PED Key secret.

Characters in Names and Passwords

Partition names

Minimum length: 1 character

Maximum length: 63 characters

Passwords

Minimum length: usually 7 characters

Maximum length: 63 characters

Partition names and Passwords

Allowed characters: !#$%'()*+,-./ 0123456789:=?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[]^_ abcdefghijklmnopqrstuvwxyz{}~

Note that 87 characters are shown above. The list includes the SPACE character. These 87 characters are the 95 normal (non-control) ASCII characters minus the 8 special characters " | & \ ; ` < >.

Do not use:

Character	Hex Value
|	0x7C
&	0x26
\	0x5C
;	0x3B
`	0x60
<	0x3C
>	0x3E

Client names

Minimum length: 1 character

Maximum length: 64 characters

Allowed characters: 0123456789ABCDEFGHIJKLMNOPQRSTUVWXY Z_abcdefghijklmnopqrstuvwxyz

Note that 63 characters are shown above. This includes the underscore character _, as well as a-z. A-Z and 0-9.

Failed Logins

HSM Admin or SO

If you fail three consecutive login attempts as HSM Security Officer, the HSM contents are rendered unrecoverable. This is a security feature (you DO have your important material backed up, don't you?) meant to thwart repeated, unauthorized attempts to access your cryptographic material. The number is ***not*** adjustable. Please note that 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 insert a wrong-color PED Key, that is not counted as a failed attempt. For a password-authenticated HSM, if you just press [Enter] with no characters, that is not counted. However, any number of characters more than zero, followed by [Enter] is counted as a bad attempt.As soon as you successfully authenticate, the counter is reset to zero.

HSM Partition Owner or User

The same security feature applies to Owner logins/activations, with some differences:

Multiple failed attempts at the user or client level affect only the HSM Partition, and not the entire Luna HSM.

Configurable

The HSM Admin (or Security Officer) can set the number of failed login attempts that trigger the feature (default is 10).

Control the Outcome

The configurable policy “SO/HSM Admin can reset User PIN” [HSM policy #15] allows you to control the outcome of too many consecutive bad authentication attempts. If the policy is “on” then the outcome is that the HSM Partition is locked out. This means that the Partition and its contents can be accessed again after the HSM Admin resets the HSM Partition Owner’s password. If the policy is “off”, then the partition is zeroized after too many bad attempts – meaning that all contents are lost and the partition must be recreated.

“Ignore failed challenge responses” can be set, which ensures that failed HSM Partition Password attempts do not cause the “failed login attempt” counter to increment.

Crypto Officer / Crypto User

If you are using the Crypto Officer / Crypto User model, the two IDs have their own independent "failed challenge response" counters. By default, each of Crypto Officer and Crypto User can make up to 10 consecutive attempts with an incorrect Password without triggering consequences on the Partition.

Forgotten Passwords

- **HSM Admin / Security Officer** – If you lose the HSM SO authentication(a password for Luna HSMs with Password Authentication; the SO PED Key for Luna HSMs with Trusted Path / PED Authentication) , you must re-initialize the HSM, which also zeroizes the HSM(the contents of the HSM become permanently unavailable, and must be replaced/regenerated after you re-initialize -- allowing anyone to change or reset the SO password without knowing the current password would not be considered good security, thus we force zeroization of all HSM contents in such a situation (either you have lost access/authentication to your own data and keys and therefore don't care that they are erased, or an attacker is attempting to gain access and you want your data and keys made unavailable, and you want to be made aware that the attack has occurred).
- **Partition Owner /Partition User / Crypto Officer** – If you lose the Partition Owner/User authentication, the HSM Admin or Security Officer can reset the password with lunacm command 'partition -resetPw'.
  The HSM Policy "21: Force user PIN change after set/reset" determines whether the Partition User can access the Partition with the password that is set by "partition -resetPw", or if the User must explicitly set a new password with "partition changePw" before being allowed to access the Partition. That policy can be used to enforce role separation between SO and User.

Resetting Passwords

HSM

There is no provision to reset the HSM Admin or SO password (for Password Authentication) or blue PED Key (Trusted Path), except by initializing the HSM (which destroys [zeroizes] 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 `lunacm 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.

The hsm init command does not require a login, and the hsm login command is not accepted if the HSM is in zeroized state.

The following are examples of the behavior of the hsm login command in various possible circumstances.

Password Authenticated HSM:

One bad login

With or without force (no difference) / interactive password:

```
Caution:  You have only TWO HSM Admin logins attempts left. If
you fail two more consecutive login attempts (i.e.
with no successful logins in between) the HSM will
be ZEROIZED!!!

  Please enter the HSM Administrators' password:
>
```

With or without force / non-interactive password:

```
>hsm login -password userpin -force

Caution:  You have only TWO HSM Admin logins attempts left. If
you fail two more consecutive login attempts (i.e.
with no successful logins in between) the HSM will
be ZEROIZED!!!

'hsm login' successful.
```

Two bad logins

Without force / interactive password:

```
Caution:  This is your LAST available HSM Admin login attempt.
If the wrong HSM Admin password is provided the HSM will
be ZEROIZED!!!

        Type 'proceed' if you are certain you have the
right login credentials or 'quit' to quit now.
> proceed

  Please enter the HSM Administrators' password:

  >
```

Without force / non-interactive password:

```
        Caution:  This is your LAST available HSM Admin login attempt.
        If the wrong HSM Admin password is provided the HSM will
        be ZEROIZED!!!

                Type 'proceed' if you are certain you have the
        right login credentials or 'quit' to quit now.
        > proceed

        'hsm login' successful.
```

With force / interactive password:

```
Caution:  This is your LAST available HSM Admin login attempt.
If the wrong HSM Admin password is provided the HSM will
be ZEROIZED!!!

  Please enter the HSM Administrators' password:
> *******

'hsm login' successful.
```

With force / non-interactive password:

```
Caution:  This is your LAST available HSM Admin login attempt.
If the wrong HSM Admin password is provided the HSM will
be ZEROIZED!!!

'hsm login' successful.
```

Trusted Path / PED Authentication (uses Luna PED and PED Keys):

One bad login

With or without force (no difference):

```
Caution:  You have only TWO HSM Admin logins attempts left. If
you fail two more consecutive login attempts (i.e.
with no successful logins in between) the HSM will
be ZEROIZED!!!

Use blue pED key?
```

Two bad logins

Without force:

```
Caution:  This is your LAST available HSM Admin login attempt.
If the wrong blue PED key is provided the HSM will
be ZEROIZED!!!

        Type 'proceed' if you are certain you have the
right login credentials or 'quit' to quit now.

        > proceed

Use blue pED key?
```

With force

```
Caution:  This is your LAST available HSM Admin login attempt.
If the wrong HSM Admin password is provided the HSM will
be ZEROIZED!!!

Use blue pED key?

'hsm login' successful.
```

Example when HSM Zeroized:

```
Error:   The HSM is zeroized due to three consecutive failures to
login as HSM Administrator.

        'hsm login' is not permitted. The HSM must be re-initialized
with the 'hsm init' command.
```

```
'hsm login' aborted.
```

Partition

If you lockout your Partition Owner / Crypto Officer with 10 bad logins AND the "SO can Reset Container PIN" policy is ON, then you MUST reset both the partition owner challenge AND the PED pin:

```
lunacm:>partition resetPw -partition Partition1
  Which part of the partition password do you wish to change?
  1. change black PED key data
  2. generate new random password for partition owner
  3. generate new random password for crypto-user
  4. both options 1 and 2
  0. abort command
  Please select one of the above options:
```

For this situation, you must choose option 4.

If the partition was activated prior to this, you must reactivate it after resetting the PED pin.

If you merely wish to change the Partition password or black PED Key, use the "partition changePw" command instead.