

SafeNet PCI-E HSM

Administration Guide

Document Information

Product Version	6.2
Document Part Number	007-011329-009
Release Date	18 December 2015

Revision History

Revision	Date	Reason
A	18 December 2015	Initial release.

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 Gemalto NV

Acknowledgements

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit.
(<http://www.openssl.org>)

This product includes cryptographic software written by Eric Young (eay@cryptsoft.com). This product includes software written by Tim Hudson (tjh@cryptsoft.com).

This product includes software developed by the University of California, Berkeley and its contributors.

This product uses Brian Gladman's AES implementation.

Refer to the End User License Agreement for more information.

Regulatory Compliance

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

USA, FCC

This device complies with Part 15 of the FCC rules. Operation is subject to the following two conditions:

- (1) This device may not cause harmful interference, and
- (2) This device must accept any interference received, including interference that may cause undesired operation.



Note: This equipment has been tested and found to comply with the limits for a “Class B” digital device, pursuant to part 15 of the FCC rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation.

If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

- Reorient or relocate the receiving antenna
- Increase the separation between the equipment and receiver
- Connect the equipment into an outlet on a circuit different from that to which the receiver is connected
- Consult the dealer or an experienced radio/TV technician for help

Changes or modifications not expressly approved by SafeNet could void the user's authority to operate the equipment.

Canada

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

Europe

This product is in conformity with the protection requirements of EC Council Directive 2004/108/EC. Conformity is declared to the following applicable standards for electro-magnetic compatibility immunity and susceptibility; CISPR22 and IEC801. This product satisfies the CLASS B limits of EN 55022.

Disclaimer

Gemalto 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, Gemalto reserves the right to revise this publication and to make changes from time to time in the content hereof without the obligation upon Gemalto 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.

Gemalto 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	Gemalto NV 4690 Millennium Drive Belcamp, Maryland 21017 USA
Email	techpubs@safenet-inc.com

CONTENTS

PREFACE	About the Administration Guide	16
Customer Release Notes		17
Gemalto Rebranding		17
Audience		17
Document Conventions		18
Notes		18
Cautions		18
Warnings		18
Command syntax and typeface conventions		18
Support Contacts		19
1	Audit Logging	21
Audit Logging Overview		21
Audit Role on HSM		21
Audit Role Available Commands		22
Audit Logging Features		22
Audit Log Secret		23
Audit Log Records		23
Audit Log Message Format		24
Synchronizing Time between HSM and Host		25
Log Secret and Log Verification		26
Capacity		26
Time Reported in Log		26
Configuration Persists		27
Audit Logging Stops Working if the Current Log File is Deleted		27
Configuring and Using Audit Logging		27
Configure Audit Logging for SafeNet PCI-E HSM or SafeNet USB HSM		28
Audit Log Operational Activities		29
Deciphering the audit log records		29
Additional Considerations		30
Audit Logging General Advice and Recommendations		30
Disk Full		31
Audit Log Categories and HSM Events		32
HSM Access		32
Log External		33
HSM Management		33
Key Management		34
Key Usage and Key First Usage		35
Audit Log Management		36
Verifying the Log Entries for Another HSM		37
Remote Audit Logging		37
UDP Considerations		37
Example using TCP		38

2 Backup and Restore HSMs and Partitions 39

Backup and Restore Overview and Best Practices	39
Backup and Restore Best Practices	40
Backup and Restore Options	40
How Partition Backup Works	41
Performing a Backup	42
Comparison of Backup Performance by Medium	42
Compatibility with Other Devices	43
Why is Backup Optional?	43
How Long Does Data Last?	43
Additional Operational Questions	44
About the SafeNet Remote Backup HSM	44
Functionality of the SafeNet Remote Backup HSM	45
Backup and Restore Options and Configurations	46
Backup HSM Installation, Storage, and Maintenance	52
Disconnecting a Backup HSM	54
Local Application-Partition Backup and Restore Using the Backup HSM	58
Partition Backup and Restore Using a Backup HSM Connected Directly to a SafeNet Network HSM Appliance	59
Partition Backup and Restore Using a Backup HSM Connected to a Local Client Workstation	63
Remote Application-Partition Backup and Restore Using the Backup HSM	68
Overview	68
Configuring the Remote Backup Service (RBS)	71
Backing Up an Application Partition to a Remotely Located Backup HSM	73
Restoring an HSM Partition From a Remotely Located Backup HSM	78
Small Form Factor Backup	82
Characteristics	82
Required Elements	83
Configuration	83
To Switch Off Small Form-Factor Backup	84
Using Small Form Factor Backup	84
Cloning and SFF Backup Option Use Cases	86
Effect on HA	89
Applicability	90
Recovering an eToken 7300 for SFF Backup	90
Output of hsm showpolicies After SFF Backup Update	97
Restoring HSM Partitions From Legacy Tokens	99
Backing Up and Restoring Your HSM SO Space	100
Troubleshooting	101
Warning: This token is not in the factory reset (zeroized) state	101

3 Capabilities and Policies 103

HSM Capabilities and Policies	103
Partition Capabilities and Policies	109

3 Backup and Restore 117

Backup & Restore Overview	117
Connect	118

Source and Target - full or partial	119
PED or Password	120
Remote Backup and Restore	120
About HSM Backup - Local and Remote	120
The Backup HSM	121
Local Backup of co-located HSMs	121
Local Backup of a Distant SafeNet HSM	122
Preparing (configuring) for Remote Backup with Remote PED	122
Remote Backup	123
Backup or SafeNet USB HSM HSM Battery Installation	124
Installing the Battery	125
Backup your HSM (non-partition) objects	127
Disconnecting SafeNet Backup HSM or SafeNet USB HSM	128
Backup your HSM Partition	128
RBS - Remote Backup Service	129
Remote Backup with RBS - the Big Picture	129
Prepare RBS to Support Backup / Restore	136
Backup your HSM Partition Remotely	138
Remote Backup	138
Assumptions	138
RBS Remote Backup with Single Remote PED on Windows	140
Restore to a SafeNet Network HSM Slot	144
Restore your HSM Partition - locally	149
Restore your HSM Partition from Token	150
Why is Backup optional?	152
How long does data last?	153
What does this mean to me?	153
What is this Error about "token not in factory reset state"?	153
Backup HSM Battery Questions	154
Should I take the battery out when storing the HSM in a safe?	155
If the battery is out, what happens?	155
If the battery dies during operation, will I lose my key material? Will corruption occur?	155
Where can I get a spare/replacement battery?	155
How do I know if the battery is dead or about to die? Can I check the status of the battery?	155
What must I do to recover function, and access to my key material, after battery removal/discharge?	156
Small Form Factor Backup	156
Characteristics	156
Required Elements	157
Configuration	157
To Switch Off Small Form-Factor Backup	158
4 Configuration File Summary	159
5 Domains	167
Single Domain Policy	167
Legacy Domains and Migration	168
6 Error Codes and Troubleshooting	172
General Troubleshooting Tips	172

Remote PED	172
System Operational and Error Messages	173
Why do I often see extra slots that say "token not present"?	173
Error: 'hsm update firmware' failed. (10A0B : LUNA_RET_OPERATION_RESTRICTED) when attempting to perform hsm update firmware?	173
KR_ECC_POINT_INVALID Error when decrypting a file encrypted from BSAFE through ECIES using ECC key with any of the curves from the x9_t2 section.	173
Slow/interrupted response from the HSM, and the "hsm show" command shows LUNA_RET_SM_SESSION_REALLOC_ERROR	174
Low Battery Message	174
Keycard and Token Return Codes	174
Library Codes	188
Vendor-defined Return Codes	194
6 High-Availability (HA) Configuration and Operation	199
High Availability (HA) Overview	199
How HA is Implemented	200
Example: Database Encryption	200
Conclusion	201
Load Balancing	202
Key Replication	203
Manual Synchronization	204
Failover	204
Reaction to Failures	205
Recovery	207
Recovery Conditions	207
Enabling and Configuring Autorecovery	208
Failure of All Members	208
Automatic Reintroduction	208
Synchronization	208
Effect of PED Operations	209
Effect of Application Restarts	209
Network failures	209
Active versus Passive Autorecovery on a SafeNet Network HSM	210
Solaris (and other Unix)	211
Performance	211
Maximizing Performance	211
Standby Members	213
Planning Your Deployment	217
HA Group Members	217
High Availability Group Sizing	218
Network Requirements	219
Upgrading and Redundancy and Rotation	219
Configuring HA	220
Set up Appliances for HA	220
Register Clients with SafeNet Network HSM HA	220
Create the HA Group	221
Verification Steps	224

HA Standby Mode [optional]	224
Using HA With Your Applications	224
HAOnly	224
Key Generation	225
Application Object Handles	225
C_FindObjects Behavior	225
Managing and Troubleshooting Your HA Groups	225
Slot Enumeration	226
Determining Which Device is in Use	226
Determining Which Devices are Active	226
Duplicate Objects	226
Adding, Removing, Replacing, or Reconnecting HA Group Members	227
Adding or Removing an HA Group Member	227
Reconnecting an Off-line Unit	227
Replacing a Failed SafeNet Network HSM	228
Replace a SafeNet Network HSM Using the same IP	230
Summary	232
Client-side - Reconfigure HA If a SafeNet Network HSM Must Be Replaced	232
Replacing the Secondary HA Group Member	236
Frequently Asked Questions	236
7 HSM Initialization	238
Initialization Overview for Password-Authenticated HSMs	238
Hard Initialization	238
Initializing	239
Logging In, Once You Have Initialized	239
Soft Initialization	239
Why choose Hard Init or Soft Init?	240
Initialization Overview for PED-authenticated HSMs	240
Hard Initialization	241
Initializing	241
Logging In, Once You Have Initialized	242
Soft Initialization	243
Why choose Hard Init or Soft Init?	243
HSM Initialization and Zeroization	243
Additional Notes	244
Re-initialize an HSM	244
Initialize an HSM With Existing Domain and Shared PED Keys	244
8 HSM Partitions	246
HSM Partitions	246
Compare Legacy Partition vs PPSO Partition	247
Authentication in General	250
Activation (PED-auth only)	251
AutoActivation (PED-auth only)	251
Partition Creation with Policy Template Using LunaCM	252
Process for a New Template	252
Modify a partition template, then apply the modified partition template	257
Delete a partition policy template	262

Separation of HSM Workspaces	263
Legacy Application Partitions	264
PPSO Application Partitions	264
Operation	264
Key Management Commands	265
Normal Usage Commands	265
Unauthenticated Commands	266
Commands That are Valid Only in a Session, But Require Special Handling	267
Configured and Registered Client Using an HSM Partition	267
Simple Troubleshooting	268
About Activation and Auto-Activation	268
Activation	268
AutoActivation	269
De-Activate a Partition	270
Removing Partitions	271
Security of Your Partition Challenge	271
How Secure Is the Challenge Secret or Password?	272
Frequently Asked Questions	272
Why do I get an error when I attempt to set the partition policies for activation (22) and auto-activation (23) on my password authenticated SafeNet Network HSM?	273
So, what is the difference in security, once Activation and Auto-activation are started?	273
9 HSM Status Values	275
10 Key Migration	278
Key Migration Procedures	278
Migrating Key Material from Older (2U) to New (1U) Appliances	278
Frequently Asked Questions	278
We want to generate keys on one HSM and copy them to other HSMs. Can they have the same object handles?	278
We want to migrate from a Microsoft Certificate Authority to a Linux CA while keeping the same private key. Does the SafeNet HSM offer any barriers to doing this?	279
11 PED Authentication	280
About the SafeNet PED	280
PED Features	280
Using the PED	281
Interaction with Other Operations	282
Versions	283
Authentication	284
Local and Remote	284
When Do I Need A PED?	285
What Do I Do?	285
Standalone or local or off-line PED operations	287
EXCEPTION: Secure Recovery	287
EXCEPTION: Remote PED	287
About PED Keys	288

Why do you need PED Keys?	288
Types of PED Key	288
What is a Set of PED Keys?	292
Physical Identification of PED Keys	294
Using PED Keys	295
Compare Password and PED Authentication	295
What is a PED PIN?	296
But what is it?	297
How to invoke/require a PED PIN with an HSM	299
Must I Use a PED PIN?	303
Should I Use a PED PIN?	303
What If I Change My Mind?	303
Does that apply to the other PED Key colors?	304
What is a Shared or Group PED Key?	304
What else do I need to know?	304
Best Practice	305
How to Use a SafeNet PED	305
SafeNet PED Keypad Functions	306
Interaction between the HSM and the PED	306
How it was - versus - how it is today	311
Restating the "obvious"?	313
Duplicating PED keys	313
Lost PED Keys or PED PINs, or passwords	315
Help! I have lost my blue/black/red/orange/purple/white PED Key or I have forgotten the password!	315
But I don't have keys or secrets in secure on-site or off-site storage! What do I do?	315
I have my PED Key, but I forgot my PED PIN! What can I do?	317
I have my PED Keys and my PED PINS, but I can't remember which one goes with which HSM (or partition)!	318
12 PED Key Management	319
PED Key Management Overview	319
"Possible" Does Not Mean "Necessary"	319
PED Keys and Operational Roles	322
Actions That Require a PED Key	324
Shared or Group PED Keys	326
How does it work?	327
The Exception	328
Domain PED Keys	328
The "New Domain" Question	328
To What Does a Domain Apply?	330
What about Legacy HSMs and Partitions?	330
Summary	331
Duplicating PED Keys	331
Considerations for Duplicate PED Keys	331
How Many PED Keys Do I Need?	333
Basic amount for operation	333
One PED Key for many HSMs - grouping or reuse	333
Many PED Keys for one HSM - MofN	336
Combining MofN with Grouped/Unique Authentication	338

Calculate PED Key requirements for some or all HSM authentication secrets	341
Maintaining Your PED Keys	342
Conclusion	342
Using MofN	343
Typical Practice	343
Common MofN Usage	343
MofN and PED PINs	344
Revoking Means Re-initializing	345
How to Add an MofN Requirement Where There Was No MofN Before	345
Implementation Suggestions	346
Make one big set, instead of three small sets	346
Complexity When Managing PED Keys	347
General Advice on PED Key Handling	347
Keep a Log	348
Apply Meaningful Labels	348
Keys	348
Updating PED Keys – Example	349
Risk of Losing access	349
PIN-change Procedure for Multiple HSMs	349
Updating PED Key for a Backup Token	353
Frequently Asked Questions	353
How should SafeNet PED Keys(*) be stored? (*Model iKey 1000 for use with SafeNet PED2)	353
So I shouldn't keep all the PED Keys for all my SafeNet HSMs in one box in a desk drawer?	354
I've lost my purple PED Key. Or, I forgot my PED PIN for my purple PED Key.	354
Do we really need to include a PED PIN with each PED Key?	355
13 Performance	356
Performance Overview	356
HA Performance	357
HSM Information Monitor	357
Notes about Monitor/Counter Behavior	357
Performance and the PE1746Enabled Setting	358
Effect on HA	358
Resetting the Internal SafeNet Network HSM PE1746Enabled Setting Following an Upgrade	358
Frequently Asked Questions	358
Can I buy a SafeNet Network HSM 1700 and later upgrade it to SafeNet Network HSM 7000?	359
Can you highlight the relative performance figures?	359
How can I achieve the kinds of performance numbers you quote?	359
Do you have any additional advice on how to interpret performance numbers? We're trying to match against a set of performance requirements that are stated as "signings per millisecond".	360
We expect to generate millions of keys per year. What is the expected number of read/write operations that your HSM memory can perform before the solid-state memory begins to fail?	360
In the "key factory" scenario, we need to generate approximately 30 ECC P224 keys/second. How many SafeNet Enterprise HSMs will we require?	360
14 Public Key Infrastructure (PKI) and Removable HSMs	361
PKI with SafeNet Network HSM	361
What to Do	361
HA	363

Using SafeNet USB HSM or Token-format HSM with SafeNet Network HSM Appliance	363
Constraints	364
PKI and HA	365
Card Reader (SafeNet DOCK 2) and Token-style HSMs	366
Frequently Asked Questions	368
We operate a Managed PKI and must satisfy our auditors that the root and intermediate keys and certs are protected according to an accepted standard, including when cloned/backed-up.	368
15 Remote PED	370
About Remote PED	370
Why do I want it?	371
How does it work?	371
One-to-One Remote PED Connections	373
Priority and Lockout	373
Remote PED Timeout	374
Ports	374
Windows 7	374
Limitations	375
Compatibility	376
Security of Remote PED	376
Remote PED Architecture	376
Remote PED and pedclient and pedserver	377
Security of Remote PED	378
Multiple HSMs and Remote PED	378
Configuring Remote PED	379
You will need:	379
Configuring the PEDClient and PEDServer	379
Relinquishing Remote PED	385
Troubleshooting	386
Using the Remote PED Feature	388
Setup Instructions	389
Troubleshooting Remote PED	395
Ped connect can fail if IP is not accessible	395
VPN	397
Timeout	397
Pedserver fails to start with "LOGGER_init failed"	397
16 Removing/Destroying Content for Safe Disposal	398
Declassify/Decommission the HSM Card	398
Resetting to Factory Condition	399
End of service and disposal	399
Needs Can Differ	399
SafeNet HSM Protects Your Keys and Objects	399
Comparison of Destruction/Denial Actions	400
RMA and Shipping Back to SafeNet	404
What Does Zeroized Mean?	404
17 User and Password Administration	406
About Changing HSM and Partition Passwords	406

HSM Passwords	407
Partition Passwords	407
Failed Logins and Forgotten Passwords	408
Appliance	408
Failed Logins	408
HSM Response When You Reach the Bad-attempt Threshold	409
Control the Outcome	410
Resetting Passwords	410
HSM	410
Partition	412
Default Challenge Password	413
18 Security Effects of Administrative Actions	414
Overt Security Actions	414
Actions with Security- and Content-affecting Outcomes	414
Elsewhere	414
Summary of Outcomes of Security-affecting Actions	415
Factory Reset HSM With Firmware <6.22.0	415
Factory Reset HSM With Firmware ≥6.22.0	415
Zeroize HSM With Firmware ≥6.22.0	416
Change Destructive HSM Policy	416
Apply Destructive CUF Update	416
HSM Initialize When Admin Not Initialized	417
HSM Initialize When Admin Initialized	417
Non-Admin Partition Initialize When the Partition is Not Initialized	418
Non-Admin Partition Initialize When the Partition is Initialized	418
19 Secure Transport Mode	419
MTK and SRK	419
Tamper and Recover with Purple Key NOT Enabled	419
Tamper and Recover with Purple PED Key Enabled	420
Behavior with Purple PED Key Enabled but MISSING or DAMAGED	421
Secure Transport Mode	421
Make a New Purple PED Key (SRK external split)	421
Master Key must be present	422
What if the purple SRK has been lost?	422
Disabling SRK	422
Compare and Contrast Some "Denial" and Destructive Scenarios	422
Secure Transport Mode [Local]	423
BACKUP	423
RECOVER	424
No Re-split?	424
Additional Notes	425
Secure Transport Mode [Remote]	426
Make a Remote PED Connection	426
Check SRK status	428
Enable SRK	428
Enter Secure Transport Mode	428
What if someone makes a new SRK while the HSM is in Transport Mode?	429

At the destination, recover from Secure Transport Mode	430
SRK key resplit	430
SRK disable	431
Re-Split Required	432
Security	432
20 Slot Numbering and Behavior	433
Order of Occurrence for Different SafeNet HSMs	433
Settings Affecting Slot Order	434
Effects of Settings on Slot List	434
Effects of New Firmware on Slot Login State	435
21 SNMP Monitoring	436
Overview and Installation	436
MIB	436
SafeNet SNMP Subagent	436
Configuration Options In the luna-snmp.conf File	437
The SafeNet Chrysalis-UTSP MIB	438
The SAFENET HSM MIB	439
SNMP Table Updates	439
hsmTable	439
hsmLicenseTable	441
hsmPolicyTable	441
hsmPartitionPolicyTable	442
hsmClientRegistrationTable	442
hsmClientPartitionAssignmentTable	442
SNMP output compared to SafeNet tools output	443
Frequently Asked Questions	446
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?	446
22 Software Maintenance and Updates	447
About Updating SafeNet Network HSM	447
How Firmware Updates Affect Agency Validation	447
Updating the Client Software	448
Updating the Appliance Software	448
Standalone Firmware Update	450
Advanced Configuration Upgrades	451
ECIES Acceleration	453
Rollback Behavior	454
Applying SafeNet HSM Capability Upgrades	455
Preparing to Upgrade	456
Installing the Upgrade Package	456
Serial Number Handling	458
23 Standards and Validations	459
About FIPS Validation	459
What does this mean to me?	460

About HSM NOT in FIPS140-2 Approved Mode	460
The FIPS-Approved Algorithms	460
What Does This Mean For Your Application?	461
What Are the Implications of Changing This Policy Setting?	461
Migrating from Non-FIPS HSM to FIPS HSM	461
SafeNet Network HSM legacy application partition	465
SafeNet Network HSM or SafeNet PCI-E HSM application partition (f/w 6.22.0 or newer) in LunaCM	477
NIST SP 800-131A: Changes to FIPS-Supported Algorithms Effective January 2014	484
Summary	484
Affected Algorithms	485
Impact on your operations	487
Mechanisms Affected	487
Other Effects	489
Modification to DES3 Algorithm for NIST Compliance	490
Common Criteria	491
Background	491
Trade-offs	491
So, What Are the Options?	492

PREFACE

About the Administration Guide

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

- "Audit Logging " on page 21
- "Backup and Restore HSMs and Partitions" on page 39
- "Capabilities and Policies" on page 103
- "Configuration File Summary" on page 159
- "Domains" on page 167
- "Error Codes and Troubleshooting" on page 172
- "High-Availability (HA) Configuration and Operation" on page 199
- "Host Trust Link Client Authentication" on page 1
- "HSM Initialization" on page 238
- "HSM Partitions" on page 246
- "HSM Status Values" on page 275
- "Key Migration" on page 278
- "PED Authentication" on page 280
- "PED Key Management" on page 319
- "Performance" on page 356
- "Public Key Infrastructure (PKI) and Removable HSMs" on page 361
- "Remote PED" on page 370
- "Removing/Destroying Content for Safe Disposal" on page 398
- "User and Password Administration" on page 406
- "Security Effects of Administrative Actions" on page 414
- "Secure Transport Mode" on page 419
- "Secure Trusted Channel (STC)" on page 1
- "Slot Numbering and Behavior" on page 433
- "SNMP Monitoring" on page 436
- "Software Maintenance and Updates" on page 447
- "Standards and Validations" on page 459

This preface also includes the following information about this document:

- "Customer Release Notes" on the next page

- "Gemalto Rebranding" below
- "Audience" below
- "Document Conventions" on the next page
- "Support Contacts" on page 19

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

Customer Release Notes

The customer release notes (CRN) provide important information about this release that is not included in the customer documentation. Read the CRN to fully understand the capabilities, limitations, and known issues for this release. You can view or download the latest version of the CRN for this release at the following location:

- http://www.securedbysafenet.com/releasenotes/luna/crn_luna_hsm_6-2.pdf

Gemalto Rebranding

In early 2015, Gemalto NV completed its acquisition of SafeNet, Inc. As part of the process of rationalizing the product portfolios between the two organizations, the Luna name has been removed from the SafeNet HSM product line, with the SafeNet name being retained. As a result, the product names for SafeNet HSMs have changed as follows:

Old product name	New product name
Luna SA HSM	SafeNet Network HSM
Luna PCI-E HSM	SafeNet PCI-E HSM
Luna G5 HSM	SafeNet USB HSM
Luna PED	SafeNet PED
Luna Client	SafeNet HSM Client
Luna Dock	SafeNet Dock
Luna Backup HSM	SafeNet Backup HSM
Luna CSP	SafeNet CSP
Luna JSP	SafeNet JSP
Luna KSP	SafeNet KSP



Note: These branding changes apply to the documentation only. The SafeNet HSM software and utilities continue to use the old names.

Audience

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

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

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

Document Conventions

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

Notes

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



Note: Take note. Contains important or helpful information.

Cautions

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



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

Warnings

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



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

Command syntax and typeface conventions

Format	Convention
bold	<p>The bold attribute is used to indicate the following:</p> <ul style="list-style-type: none"> • Command-line commands and options (Type <code>dir /p</code>.) • Button names (Click Save As.) • Check box and radio button names (Select the Print Duplex check box.) • Dialog box titles (On the Protect Document dialog box, click Yes.) • Field names (User Name: Enter the name of the user.) • Menu names (On the File menu, click Save.) (Click Menu > Go To > Folders.)

Format	Convention
	<ul style="list-style-type: none"> User input (In the Date box, type April 1.)
<i>italics</i>	In type, the italic attribute is used for emphasis or to indicate a related document. (See the <i>Installation Guide</i> 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> <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> <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

Contact method	Contact
Address	Gemalto NV 4690 Millennium Drive Belcamp, Maryland 21017 USA

Contact method	Contact	
Phone	Global	+1 410-931-7520
	Australia	1800.020.183
	China	(86) 10 8851 9191
	France	0825 341000
	Germany	01803 7246269
	India	000.800.100.4290
	Netherlands	0800.022.2996
	New Zealand	0800.440.359
	Portugal	800.1302.029
	Singapore	800.863.499
	Spain	900.938.717
	Sweden	020.791.028
	Switzerland	0800.564.849
	United Kingdom	0800.056.3158
	United States	(800) 545-6608
Web	www.safenet-inc.com	
Support and Downloads	www.safenet-inc.com/support Provides access to the Gemalto Knowledge Base and quick downloads for various products.	
Technical Support Customer Portal	https://serviceportal.safenet-inc.com Existing customers with a Technical Support Customer Portal account can log in to manage incidents, get the latest software upgrades, and access the Gemalto Knowledge Base.	

Audit Logging

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

- "Audit Logging Overview" below
- "Configuring and Using Audit Logging" on page 27
- "Audit Logging General Advice and Recommendations" on page 30
- "Audit Log Categories and HSM Events" on page 32
- "Verifying the Log Entries for Another HSM" on page 37
- "Remote Audit Logging " on page 37

Audit Logging Overview

Beginning with release 5.2, SafeNet HSMs consolidate and enhance auditing of HSM operations. The audit logging feature works only with hardware and software update levels at, or newer than, the versions that introduced the feature:

Hardware

- SafeNet PCI-E HSM 5.x

Software

- Client Software 5.2.0 or newer
- HSM Firmware 6.10.1 or newer
- SafeNet PED 2.5.0-2 or newer (if PED-authenticated HSM)

Audit Role on HSM

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

Password-authenticated HSMs

For SafeNet HSMs with Password Authentication, the auditor role logs into the HSM to perform their activities using a password.

PED-authenticated HSMs

For SafeNet HSMs with PED Authentication, the auditor role logs into the HSM to perform their activities using the Audit (white) PED Key. The Audit feature works only with SafeNet PED version 2.5.0-3 or newer. Older versions of PED firmware are not aware of the Audit role and Audit Key.

Role Initialization

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

Audit Role Available Commands

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

```
lunacm:>audit
```

The following sub commands are available:

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

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

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

Audit Log Secret

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

Audit Log Records

A log record consists of two fields – the log message and the HMAC for the previous record. When the HSM creates a log record, it uses the log secret to compute the SHA256-HMAC of all data contained in that log message, plus the HMAC of the previous log entry. The HMAC is stored in HSM flash memory. The log message is then transmitted,

along with the HMAC of the previous record, to the host. The host has a logging daemon to receive and store the log data on the host hard drive.

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

MSG 1	HMAC 0
	...
MSG n-1	HMAC n-2
MSG n	HMAC n-1
...	
MSG n+m	HMAC n+m-1
MSG n+m+1	HMAC n+m
...	
MSG end	HMAC n+m-1
Recent HMAC in NVRAM	HMAC end

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

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

Audit Log Message Format

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

Offset	Length (Chars)	Description
0	10	Sequence number
10	1	Comma
11	17	Timestamp

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

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

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

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

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

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



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

Capacity

The log capacity of SafeNet HSMs varies depending upon the physical memory available on the device. The SafeNet PCI-E HSM and the HSM contained in the SafeNet Network HSM appliance are the SafeNet K6 HSM card. The HSM inside both the SafeNet USB HSM and the SafeNet 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 a SafeNet Network HSM 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 decommission 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 might lose log entries.

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

Configuring and Using Audit Logging

The overall sequence when initializing an HSM that will use Security Audit Logging is as follows:

1. **Configure the SafeNet Network HSM appliance or the SafeNet PCI-E HSM/SafeNet USB HSM host workstation to use the network time protocol (NTP).**

Configure access to at least two geographically separated NTP servers for redundancy. Select at least one NTP server that is known to have a high degree of accuracy and reliability (servers associated with national standards bodies are good candidates) as one of the configured servers.

2. **Initialize the Audit Officer role.**

This enables logging for all subsequent actions performed by the SO and partition User(s).

3. **Execute the 'audit sync' command.**

This ensures that the HSM's clock is synchronized with the host time (which should also be synchronized with the NTP server) and that all subsequent log records will have a valid and accurate timestamp.

4. **Configure the audit category and level of audit**

You can specify the level of audit appropriate for needs of the organization's policy and the nature of the application(s) using the HSM. Security audits can generate a very large amount of data, which consumes HSM processing resources, host storage resources, and makes the job of the Audit Officer quite difficult when it comes time to review the logs. For this reason, ensure that you configure audit logging such that you capture only relevant data, and no more.

For example, the 'First Key Usage Only' category is intended to assist Audit Officers to capture the relevant data in a space-efficient manner for high processing volume applications. On the other hand, a top-level Certificate Authority would likely be required, by policy, to capture all operations performed on the HSM but, since it is typically

not an application that would see high volumes, configuring the HSM to audit all events would not impose a significant space and/or performance premium in that situation.

5. **Configure log rotation and remote logging server(s) as necessary.**

The settings for these configuration elements will often be dictated by the organization's Audit and/or IT policies and procedures. As with configuring the audit category, the Audit Officer should be prudent in making these configuration settings. It is recommended that the default setting of 'Rotate Log Daily' be maintained until the typical/average logging rate can be determined. The use of redundant remote log servers, accessible only by the members of the audit team, is strongly recommended.

6. **Initialize the HSM and create partitions as necessary.**

At this point, the HSM is ready to be turned over to the SO to initialize it and begin creating the partitions needed to serve the processing applications.

Configure Audit Logging for SafeNet PCI-E HSM or SafeNet USB HSM

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

Required SafeNet HSM 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.

To configure audit logging

1. Set the slot focus to the HSM administrative partition of the desired HSM:

```
lunacm:>slot set slot <number>.
```

2. Initialize the **Auditor** role:

```
lunash:>role init -name Auditor
```

- On password-authenticated HSMs, you are prompted for a domain string and password
- On PED-authenticated HSMs, you are referred to SafeNet PED, which prompts for a white PED Key.

3. Now that the **Auditor** role exists on the HSM, the auditing function must be configured. However, before you can configure you must log in as the **Auditor** user:

```
lunacm:>role login -name Auditor
```

- On password-authenticated HSMs, you are prompted to enter the password for the **Auditor** user.
- On PED-authenticated HSMs, you are referred to SafeNet PED, which prompts for the white PED Key for the **Auditor** user.

4. Configure audit logging:

```
lunash:> audit config
```



Note: The first time you configure audit logging, we suggest using only the "?" option, in order to see all the available options in the configuration process.

Log Entries

Log entries are made within the HSM, and are written to the currently active log file on the appliance file system. When a log file reaches the rotation trigger, it is closed, and a new file gets the next log entry. The number log files on the appliance 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 HSM, if you wish to free the space.

For USB-connected external (SafeNet USB HSM) and PCI-bus internal HSMs (SafeNet PCI-E HSM), the lunacm tool includes options to set the log file size and the log file path (which are then written to the config file).

Audit Log Operational Activities

lunacm:> **audit export**

- 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
```

- On the computer where the HSM is attached, that you will use to verify the downloaded audit log file, run:

```
/usr/safenet/lunaclient/bin :>scp audit@mylunasa1:151170.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 SafeNet Network HSM's file system (in the "audit" account) and stores the copy on your local computer file system in the directory from which you issued the command.

- Launch LunaCM,

```
/usr/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 SafeNet Network HSM.

- Import the Audit Logging secret into the locally attached HSM:

```
lunacm:>audit import file 151170.lws
```

- Verify the file:

```
lunacm:>audit verify file mylunsa1_audit_2014-02-28.tgz
```

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

Deciphering the audit log records

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

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

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

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

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

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

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

```
150718020
```

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

Additional Considerations

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

Audit Logging General Advice and Recommendations

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

- maintaining an audit trail that can later be used to reconstruct a particular action or set of actions (i.e., forensics);
- maintaining an audit trail that can later be used to trace the actions of an application or individual user (i.e., accounting); and
- maintaining an audit trail that can later be used to hold a specific individual accountable for his/her actions (i.e., non-repudiation)

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

- Use a shell script to execute the audit sync command at least once every 24 hours, provided the host has maintained its connection(s) to its configured NTP server(s).
- Do not allow synchronization with the host's clock if the host has lost connectivity to NTP. This ensures that the

HSM's internal clock is not set to a less accurate time than it has maintained internally. In general, the HSM's RTC will drift much less than the host's RTC and will, therefore, be significantly more accurate than the host in the absence of NTP.

- Review logs at least daily and adjust configuration settings if necessary. It is important that any anomalies be identified as soon as possible and that the logging configuration that has been set is effective. If possible, use the remote logging feature to transmit log data to a Security Information and Event Management (SIEM) system to automatically analyze log data and identify anomalous events.
- In the case of SafeNet Network HSM, execute the audit log tarlogs lush command regularly to archive the audit logs and transfer them to a separate machine for long term storage. Also, execute the 'audit log clear' lush command regularly to free up the audit log disk space on SafeNet Network HSM.
- Consider installing and configuring an HSM (e.g., SafeNet USB HSM or PCI) connected to the remote log server to act as a "verification engine" for the remote log server. Ensure that the log secret for the operational HSM(s) has been shared with the log server verification HSM.

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

- The audit log records are comma-delimited. We recommend that full use be made of the CSV formatting to import records into a database system or spreadsheet tool for analysis, if an SIEM system is not available.
- The ASCII hex data representing the command and returned values and error code should be examined if an anomaly is detected in log review/analysis. It may be possible to match this data to the HSM's dual-port data. The dual-port, if it is available, will contain additional data that could be helpful in establishing the context surrounding the anomalous event. For example, if an unexpected error occurs it could be possible to identify the trace through the firmware subsystems associated with the error condition. This information would be needed to help in determining if the error was unexpected but legitimate or if it was forced in an attempt to exploit a potential weakness (e.g., searching for buffer overflows).

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

Disk Full

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

To resolve that situation, the audit user must:

- archive the audit logs on the host side,
- move them to some other location for safe storage,
- clear the audit log directory, and
- restart the logger daemon.

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

Audit Log Categories and HSM Events

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

HSM Access

HSM Event	Description
LUNA_LOGIN	C_Login. This event must be allowed to proceed even if the result should be logged but cannot (for example, due to a log full condition).
LUNA_LOGOUT	C_Logout. This event must be allowed to proceed even if the result should be logged but cannot (for example, due to a log full condition).
LUNA_MODIFY_OBJECT	C_SetAttributeValue
LUNA_OPEN_SESSION	C_OpenSession. This event must be allowed to proceed even if the result should be logged but cannot (for example, due to a log full condition).
LUNA_CLOSE_ALL_SESSIONS	C_CloseAllSessions
LUNA_CLOSE_SESSION	C_CloseSession This event must be allowed to proceed even if the result should be logged but cannot (for example, due to a log full condition).
LUNA_OPEN_ACCESS	CA_OpenApplicationID
LUNA_CLEAN_ACCESS	CA_Restart, CA_RestartForContainer
LUNA_CLOSE_ACCESS	CA_CloseApplicationID
LUNA_LOAD_CUSTOM_MODULE	CA_LoadModule
LUNA_LOAD_ENCRYPTED_CUSTOM_MODULE	CA_LoadEncryptedModule
LUNA_UNLOAD_CUSTOM_MODULE	CA_UnloadModule
LUNA_EXECUTE_CUSTOM_COMMAND	CA_PerformModuleCall
LUNA_HA_LOGIN	CA_HAGetLoginChallenge, CA_HAAnswerLoginChallenge, CA_HALogin, CA_HAAnswerMofNChallenge, HAActivateMofN

Log External

HSM Event	Description
LUNA_LOG_EXTERNAL	CA_LogExternal

HSM Management

HSM Event	Description
LUNA_ZEROIZE	CA_FactoryReset This event is logged unconditionally.
LUNA_INIT_TOKEN	C_InitToken This event is logged unconditionally.
LUNA_SET_PIN	C_SetPIN
LUNA_INIT_PIN	C_InitPIN
LUNA_CREATE_CONTAINER	CA_CreateContainer
LUNA_DELETE_CONTAINER	CA_DeleteContainer, CA_DeleteContainerWithHandle
LUNA_SEED_RANDOM	C_SeedRandom
LUNA_EXTRACT_CONTEXTS	C_GetOperationState
LUNA_INSERT_CONTEXTS	C_SetOperationState
LUNA_SELF_TEST	C_PerformSelfTest
LUNA_LOAD_CERT	CA_SetTokenCertificateSignature
LUNA_HA_INIT	CA_HAInit
LUNA_SET_HSM_POLICY	CA_SetHSMPolicy
LUNA_SET_DESTRUCTIVE_HSM_POLICY	CA_SetDestructiveHSMPolicy
LUNA_SET_CONTAINER_POLICY	CA_SetContainerPolicy
LUNA_SET_CAPABILITY	Internal, for capability update
LUNA_CREATE_LOGIN_CHALLENGE	CA_CreateLoginChallenge
LUNA_REQUEST_CHALLENGE	CA_SIMInsert, CA_SIMMultiSign
LUNA_PED_INIT_RPV	CA_InitializeRemotePEDVector
LUNA_PED_DELETE_RPV	CA_DeleteRemotePEDVector
LUNA_MTK_LOCK	Internal, for manufacturing

HSM Event	Description
LUNA_MTK_UNLOCK_CHALLENGE	Internal, for manufacturing
LUNA_MTK_UNLOCK_RESPONSE	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
LUNA_FW_UPGRADE_FINAL	CA_FirmwareUpdate
LUNA_FW_ROLLBACK	CA_FirmwareRollback
LUNA_MTK_SET_STORAGE	CA_MTKSetStorage
LUNA_SET_CONTAINER_SIZE	CA_SetContainerSize

Key Management

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

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

Key Usage and Key First Usage

HSM Event	Description
LUNA_ENCRYPT_INIT	C_EncryptInit
LUNA_ENCRYPT	C_Encrypt
LUNA_ENCRYPT_END	C_EncryptFinal
LUNA_DECRYPT_INIT	C_DecryptInit
LUNA_DECRYPT	C_Decrypt
LUNA_DECRYPT_END	C_DecryptFinal
LUNA_DIGEST_INIT	C_DigestInit
LUNA_DIGEST	C_Digest
LUNA_DIGEST_KEY	C_DigestKey
LUNA_DIGEST_END	C_DigestFinal
LUNA_SIGN_INIT	C_SignInit

HSM Event	Description
LUNA_SIGN	C_Sign
LUNA_SIGN_END	C_SignFinal
LUNA_VERIFY_INIT	C_VerifyInit
LUNA_VERIFY	C_Verify
LUNA_VERIFY_END	C_VerifyFinal
LUNA_SIGN_SINGLEPART	C_Sign
LUNA_VERIFY_SINGLEPART	C_Verify
LUNA_WRAP_CSP	CA_CloneMofN_Common
LUNA_M_OF_N_DUPLICATE	CA_DuplicateMofN
LUNA_ENCRYPT_SINGLEPART	C_Encrypt
LUNA_DECRYPT_SINGLEPART	C_Decrypt
LUNA_PE1746_COMMAND	Used when PE1746 is enabled

Audit Log Management

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

Verifying the Log Entries for Another HSM

You can use one HSM to verify the audit log files/entries that were created by another HSM. You can only verify the logs that are stored in the **ready_for_archive** folder; you cannot verify log files that are currently being written.

To verify the log entries for another HSM

1. Export the secret on SafeNet HSM1 (`audit secret export`)
2. Tar logs on SafeNet HSM1 host (`audit log tar`)
3. Transfer the secret to SafeNet HSM2 (`scp`)
4. Transfer the archive to SafeNet HSM2 (`scp`)
5. Import the secret onto SafeNet HSM2 (`audit secret import -f <SafeNetHSM1_SN>.lws -serialtarget <SafeNetHSM2_SN> -serialsource <SafeNetHSM1_SN>`)



Note: If you are verifying logs on a different HSM, you must provide the **serialsource** argument, as the SafeNet HSM will not look for other SafeNet HSM log files without it.

6. Untar logs on SafeNet HSM2 (`audit log untarlogs -f audit-<SafeNetHSM1_SN>.tgz`)
7. Verify log file. (`audit log verify -f <LOG_FILENAME>.log -serialtarget <SafeNetHSM2_SN> -serialsource <SafeNetHSM1_SN>`)



Note: You cannot pass in the full path to the log file on SafeNetSA, as the command does not parse the slashes, but it will look in all the subfolders under the HSM serial number that you specified with serialsource.

Remote Audit Logging

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



Note: You or your network administrator will need to adjust your firewall to pass this traffic (iptables).

UDP Considerations

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

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

Possible approaches include the following:

- With templates:

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

- Without templates:

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

- **Dynamic filename:**

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



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

Example using TCP

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

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

```
lunash:> audit remotehost add -host 172.20.9.160 -protocol tcp -port 1660
```

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

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

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

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

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

```
# service rsyslog restart
```

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

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

Backup and Restore HSMs and Partitions

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

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

- "Backup and Restore Overview and Best Practices " below
- "About the SafeNet Remote Backup HSM" on page 44
- "Backup HSM Installation, Storage, and Maintenance" on page 52
- "Local Application-Partition Backup and Restore Using the Backup HSM" on page 58
- "Remote Application-Partition Backup and Restore Using the Backup HSM" on page 68
- "Small Form Factor Backup" on page 156
- "Restoring HSM Partitions From Legacy Tokens" on page 99
- "Backing Up and Restoring Your HSM SO Space" on page 100
- "Troubleshooting" on page 101

Backup and Restore Overview and Best Practices

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

- "Backup and Restore Best Practices" on the next page
- "Backup and Restore Options" on the next page
- "How Partition Backup Works" on page 41
- "Performing a Backup" on page 42
- "Comparison of Backup Performance by Medium" on page 42
- "Compatibility with Other Devices" on page 43
- "Why is Backup Optional?" on page 43
- "How Long Does Data Last?" on page 43
- "Additional Operational Questions" on page 44

Backup and Restore Best Practices

In order to ensure that your data is protected in the event of a failure or other catastrophic event, Gemalto recommends that you adhere to the following best practices as part of a comprehensive backup strategy:

- Develop and document a backup and recovery plan. This plan should include the following:
 - what is being backed up.
 - the backup frequency.
 - where the backups are stored.
 - who is able to perform backup and restore operations.
 - frequency of exercising the recovery test plan.
- Make multiple backups. To ensure that your backups are always available, build redundancy into your backup procedures.
- Use off-site storage. In the event of a local catastrophe, such as a flood or fire, you might lose both your working HSMs and locally stored backup HSMs. To fully protect against such events, always store a copy of your backups at a remote location. You can automate off-site backups using the remote backup feature, See "[Remote Application-Partition Backup and Restore Using the Backup HSM](#)" on page 68 for more information.
- Regularly exercise your disaster recovery plan. Execute your recovery plan at least semi-annually (every six months) to ensure that you can fully recover your key material. This involves retrieving your stored Backup HSMs and/or SFF tokens and restoring their contents to a test partition, to ensure that the data is intact and that your recovery plan works as documented.



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

Backup and Restore Options

The available options for backing up your SafeNet PCI-E HSM and SafeNet Network HSM partitions include:

- Local or remote backup to a SafeNet Remote Backup HSM (see "[Local Application-Partition Backup and Restore Using the Backup HSM](#)" on page 58 and "[Remote Application-Partition Backup and Restore Using the Backup HSM](#)" on page 68)
- Backup to a SafeNet eToken 7300 small form factor (SFF) USB backup token, through a SafeNet PED (see "[Small Form Factor Backup](#)" on page 156)
- Key synchronization among two or more SafeNet HSMs in an HA configuration (see "[High-Availability \(HA\) Configuration and Operation](#)" on page 199)
- Direct cloning between two HSMs locally connected to one host
- Any combination of the above methods, to suit your needs.

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

How Partition Backup Works

HSM partition backup securely clones Partition objects from a named HSM partition, to a SafeNet Remote Backup HSM (supports remote or local backups) or to a Small Form Factor USB Backup (SafeNet eToken 7300) plugged into a SafeNet PED. This allows you to safely and securely preserve important keys, certificates, etc., away from the primary SafeNet HSM. It also allows you to restore the backup device's contents onto more than one HSM partition, if you wish to have multiple partitions with identical contents.

In order to back up a partition you must own it and be able to see it. This means that you can use LunaSH to back up any partitions you own on a SafeNet Network HSM appliance, or LunaCM to backup any SafeNet PCI-E HSM or SafeNet USB HSM or SafeNet Network HSM partitions that are visible as slots.

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

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

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

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

Backup Devices

You can back up all of your partitions to a SafeNet Remote Backup HSM, or a single partition to a SafeNet eToken 7300 (SFF token):

SafeNet Remote Backup HSM (Backup HSM)



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

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

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

SafeNet USB HSM

You can backup an application partition in SafeNet PCI-E HSM or a SafeNet USB HSM to another SafeNet USB HSM that is directly connected to the same host, by means of the command "[partition clone](#)" on [page 1](#) between the respective slots.

SafeNet eToken 7300 (SFF token)

The SafeNet eToken 7300 is a small form factor USB Backup Device that connects to the PED Key port on a SafeNet PED, which connects to the HSM remotely, via Remote PED connection. The Small Form Factor USB Backup is described in "[Small Form Factor Backup](#)" on [page 156](#). SFF backups are supported in LunaCM only.

Performing a Backup

To perform a backup, you identify the partition to be backed up (source), and the partition that will be created (or added to) on the Backup HSM or SFF eToken - the Token Partition Name - and then specify whether to **add** only unique objects (objects that have not previously been saved onto the target partition), or to completely **replace** the target partition (overwrite it).

LunaCM

If you are using lunacm:> on a workstation, the command is:

```
lunacm:> partition archive backup -slot <slot> -pas <password> -par <backup partition>
or
lunacm:> partition archive backup -slot <etoken> -label <label_for_SFF_token>
```

If backing up to an SFF device, you can optionally specify a comma-delimited list of handles for specific objects to backup; otherwise, the default is to backup all objects in the source partition.

The lunacm:> version assumes that the target partition already exists with the appropriate domain, while the lunash:> version expects you to provide the domain, or prompts for it if it is not provided (for Password-authenticated HSMs).

Replacing or Appending

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

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

Comparison of Backup Performance by Medium

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

Backup Destination	Time Required for Operation	Comment
SafeNet Backup HSM (PW-auth), local	5 seconds	Password is supplied with the command

Backup Destination	Time Required for Operation	Comment
SafeNet Backup HSM (PED-auth), local	5 seconds plus...	Add any time required for PED-key operations
SafeNet eToken 7300 (SFF device) connected to Remote PED on workstation, distant from SafeNet Network HSM	7 minutes 13 seconds	Approximately 130 seconds is initialization of the SafeNet eToken 7300 device; remainder is secure copying of partition objects

Compatibility with Other Devices

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

Why is Backup Optional?

In general, a SafeNet HSM or HSM partition is capable of being backed up to a SafeNet Backup HSM or SafeNet Small Form-Factor Backup eToken. The backup capability is considered a good and desirable and necessary thing for keys that carry a high cost to replace, such as Certificate Authority root keys and root certificates.

However, backup devices are optional equipment for SafeNet HSMs. There are at least three reasons for this:

1. Some customers don't care. They may be using (for example) SSL within a controlled boundary like a corporation, where it is not a problem to simply tell all employees to be prepared to trust a new certificate, in the event that the previous one is lost or compromised. In fact it might be company policy to periodically jettison old certificates and distribute fresh ones. Other customers might be using software that manages lost profiles, making it straightforward to resume work with a new key or cert. The certificate authority that issued the certificates would need backup, but the individual customers of that certificate authority would not. In summary, it might not be worthwhile to backup keys that are low-cost (from an implementation point of view) to replace. Keys that carry a high cost to replace should be backed up.
2. SIM (Secure Identity Management or Multi-Million Keys) does not co-exist with standard SafeNet cloning function. The SIM Master Secret Key on the HSM can be backed up, but HSM Partitions are not used in the SIM configuration, so there are no contents to backup.
3. Some countries do not permit copying of private keys. If you are subject to such laws, and wish to store encrypted material for later retrieval (perhaps archives of highly sensitive files), then you would use symmetric keys, rather than a private/public keypair, for safe and legal backup.

How Long Does Data Last?

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

No electronic storage is forever. If your SafeNet HSM is operated within an ambient temperature range of 0 degrees Celsius to +40 degrees Celsius, or stored between -20 degrees Celsius and +65 degrees Celsius, then (according to industry-standard testing and estimation methods) your data should be retrievable for twenty years from the time that

the token was shipped from the factory. This is a conservative estimate, based on worst-case characteristics of the system components.

Additional Operational Questions

Is SafeNet Remote Backup HSM capable of backing up multiple SafeNet HSMs or is it a one-to-one relationship?

For example, if we had two SafeNet Network HSM appliances each with two partitions, or if we had four SafeNet PCI-E HSMs, could we backup all four partitions to a single Backup HSM? If yes, do they need to be under the same domain?

Answer

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

Can a SafeNet Remote Backup HSM keep multiple backups of a single partition?

For example, could we perform a backup of an application partition one month and then back it up again next month without overwriting the previous month?

Answer

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

Is Small Form Factor USB Backup capable of backing up multiple SafeNet HSMs or partitions?

Answer

One Small Form Factor USB Backup device can back up one SafeNet HSM partition. Backup operations are overwrites, not incremental or additive. To use the same Small Form Factor USB Backup device to backup a different HSM partition, you lose any data or objects that were already on the SFF Backup device.

About the SafeNet Remote Backup HSM

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

- "Functionality of the SafeNet Remote Backup HSM" on the next page
- "Backup and Restore Options and Configurations" on page 46



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

Functionality of the SafeNet Remote Backup HSM

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

Storage Capacity and Supported Number of Partitions

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

The base configuration for SafeNet Backup HSM is 20 partitions and 15.5 Mb of space, allowing you to backup a SafeNet Network HSM with up to twenty partitions, or any combination of partitions on individual SafeNet HSMs, up to the maximum memory available on the Backup HSM. SafeNet Network HSM at firmware 6.22.0 or newer can be updated via capability update to support 50 or 100 partitions. You have the option of purchasing and adding capability upgrades for 50 or 100 partitions to SafeNet Network HSM, as well as to the SafeNet Backup HSM.



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

Upgrading the Number of Supported Partitions

The 50 Partitions Capability Upgrade and the 100 Partitions Capability Upgrade are provided in the form of CUFs (capability update files) that can be applied to a SafeNet Backup HSM connected to your workstation, in the same fashion as upgrades are applied to an installed SafeNet PCI-E HSM or to a USB-connected SafeNet USB HSM.

The 50 Partitions Capability Upgrade and the 100 Partitions Capability Upgrade are provided in the form of a secure package (.spkg file) that can be uploaded (via scp or pscp) for processing by SafeNet Network HSM to upgrade SafeNet Network HSM partition limit, or to upgrade the partition limit of a SafeNet Backup HSM connected directly to the SafeNet Network HSM appliance for local backup.

When your SafeNet Backup HSM is connected locally to a SafeNet Network HSM appliance, use the upgrade instructions at "[Applying SafeNet HSM Capability Upgrades](#)" on page 455 to apply an upgrade to increase the number of HSM partitions that can be backed up to the device.

Domains and Backups

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

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

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

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

- Full/replace backup or restore creates a new target partition with the same domain as the source partition.
- Partial (additive/incremental) backup or restore requires the existing source and target partitions to have the same domain before the operation can start.

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

PED or Password Authentication

The Backup HSM creates a partition with matching authentication type to the SafeNet HSM partition that is being backed up. That does not work in the opposite direction, however. The Backup HSM can restore a partition (or contents of a partition) only to a SafeNet HSM of matching authentication type.

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

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

Backup and Restore Options and Configurations

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

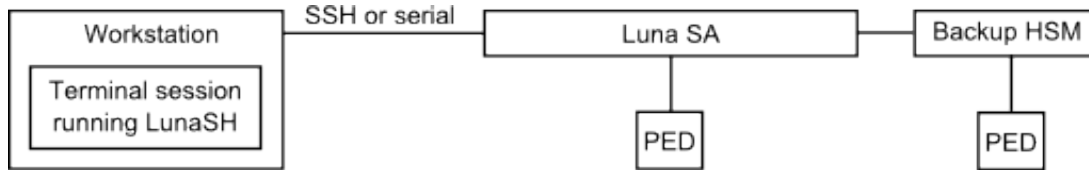
- **Local backup of any SafeNet HSM**, where all components are co-located. This is a possible scenario with all SafeNet HSMs, but is more likely with direct-connect, local-to-the-client HSMs such as SafeNet PCI-E HSM. It is unlikely for SafeNet Network HSM, simply because SafeNet Network HSM normally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

In local mode, you connect the Backup HSM directly, via USB, to a SafeNet Network HSM appliance or SafeNet PCI-E HSM host server. That is, local backup is local to the HSM being backed-up, not necessarily local to the administrator who is directing the process, who might be far away.

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

Backing Up a Local HSM to a Directly Connected Backup HSM

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

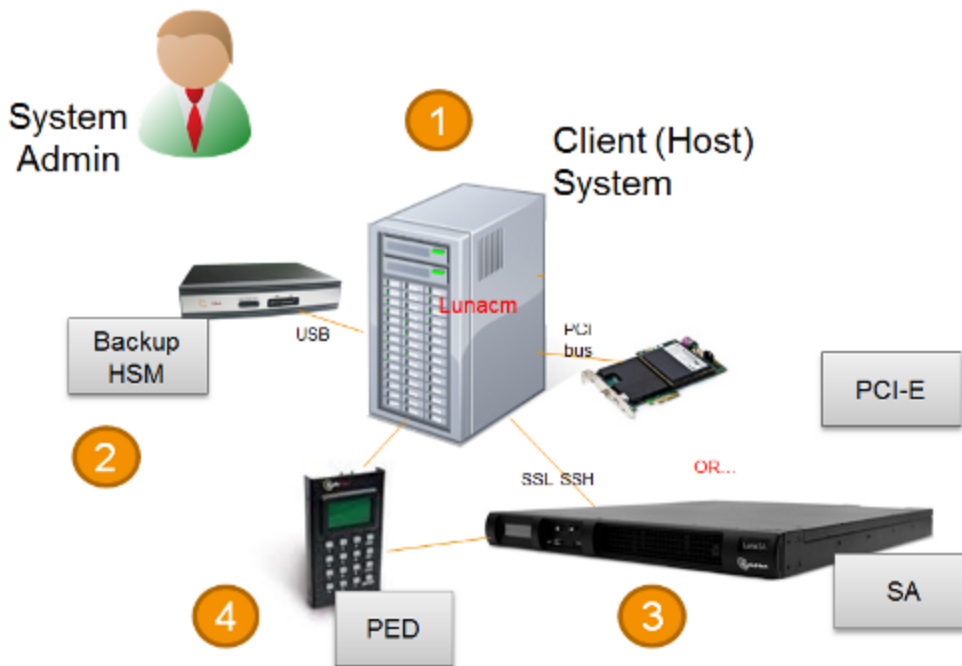


The workstation is simply a display terminal for LunaSH running on the appliance. It does not require the SafeNet client software.

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

Backup to a Backup HSM Connected to a Local Client

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



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

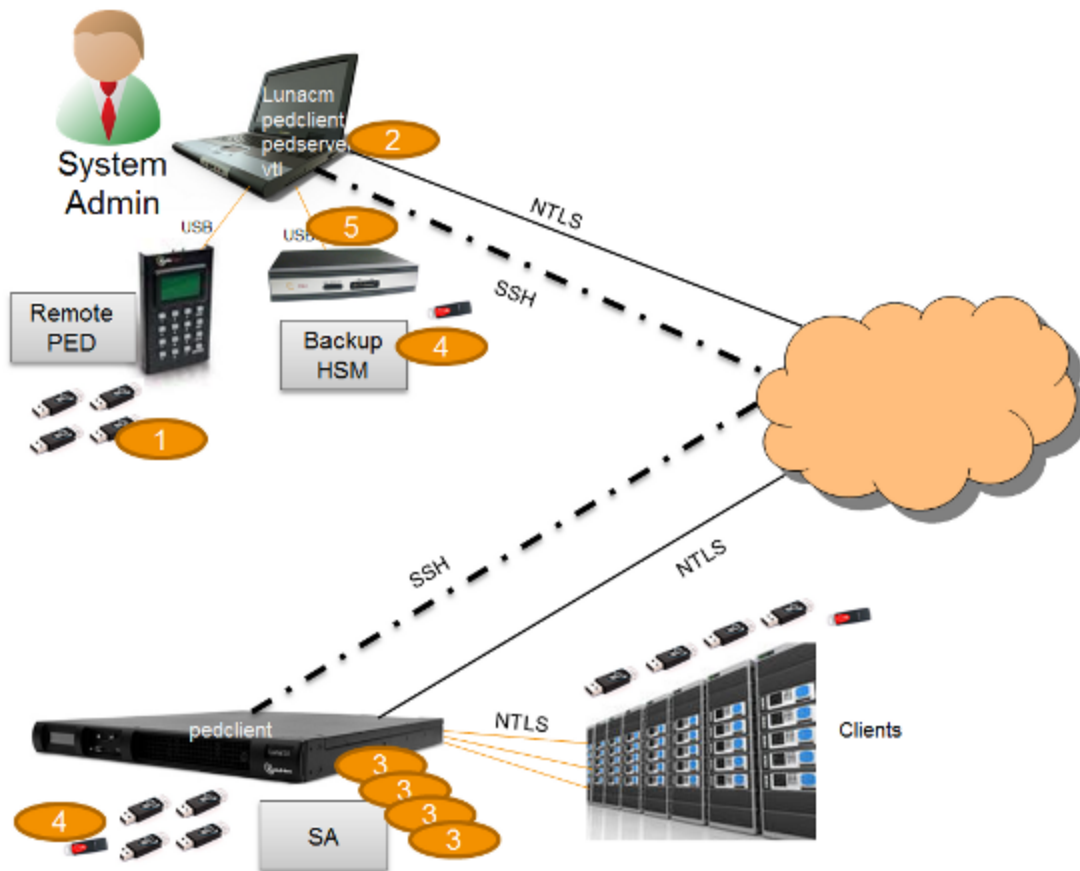
The other two backup and restore options, local backup of a distant SafeNet Network HSM and remote backup of any SafeNet HSM require that PED operations be performed remotely. For that reason, HSMs must be prepared (locally) in advance by having orange Remote PED Keys created and matched with each HSM.

Backing Up a Remote HSM to a Locally Connected Backup HSM

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

- the SafeNet HSM client software with the Remote PED options is installed.
- a Remote SafeNet PED is connected.
- the SafeNet Backup HSM is connected.

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



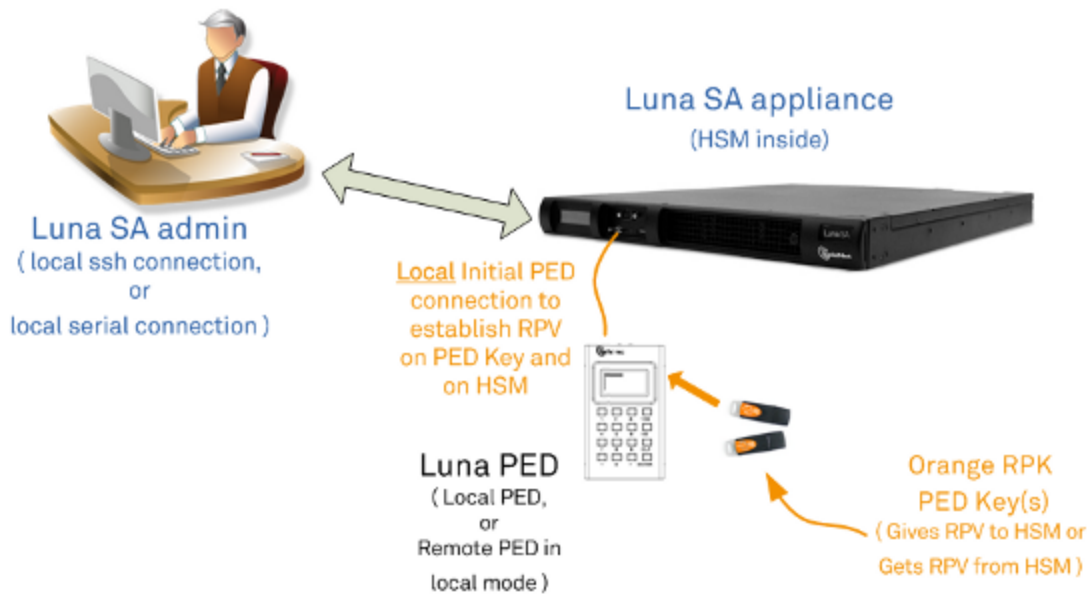
- | | |
|---|---|
| 1 | The admin must have client access to each partition being backed up. In this scenario, the admin must have black PED Keys and passwords for the partitions. |
|---|---|

2	The local host is used to control the backup/restore. The SafeNet HSM client vtl software is used to generate and trade certificates with SafeNet Network HSM, to create an NTLS link. The SafeNet PEDServer software running on the local host, in conjunction with the PEDClient software running on the SafeNet Network HSM, provides remote PED access to the SafeNet Network HSM.
3	The local host can see the SafeNet Network HSM partitions as slots in LunaCM. The SafeNet PEDClient software runs on the SafeNet Network HSM when it needs to access the Remote PED via the SafeNet PEDServer software running on the local host.
4	Every slot on the Backup HSM must have same domain (red PED Key) as the matching slot on the working HSM. The domain (red) PED Keys can be different for each partition or they can share one common domain, re-used for all partitions. The important consideration is that whatever domain situation exists on the primary HSM must be matched on the Backup HSM.
5	The local host can see the Backup HSM as a slot in LunaCM. Because the local host views the backup/restore operation in this scenario as a local transaction, between two slots visible to LunaCM on the local host, the remote backup service (RBS) is not needed.

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

Backing Up a Remote HSM to a Remotely Connected Backup HSM

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

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

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

- one to run the ssh administrative connection to the shell (lunash:>) on the SafeNet Network HSM appliance
- one to run the Remote PED server, with the SafeNet PED2 (in remote mode) connected via USB to the computer and separately connected to the mains electrical power source
- one to run a client session with vtl and the SafeNet Remote Backup driver, and with the SafeNet Remote Backup HSM with its own local SafeNet PED attached

As noted previously, the orange PED Keys contain a Remote PED Vector (RPV) that matches the RPV inside the SafeNet Network HSM. It is the presence of that RPV at both ends that allows the connection to be made between the HSM and the Remote PED. At the same time, the SafeNet Network HSM and the SafeNet Remote Backup HSM must share the same cloning domain, in order for backup and restore (cloning) operations to take place between the two HSMs. Therefore, red PED Keys with that cloning domain must be available.

As of SafeNet HSM 5.2, SafeNet HSMs use Remote Backup Service (RBS) to facilitate Remote Backup. Where formerly we ran the remote backup from the "vtl" utility, we now use vtl only for the certificate exchange that makes a computer a client of a distant SafeNet Network HSM partition.

Required Software

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

PedClient is needed on any host that must reach out to a pedserver instance and a Remote PED. PedClient instances can also communicate with each other to facilitate RBS

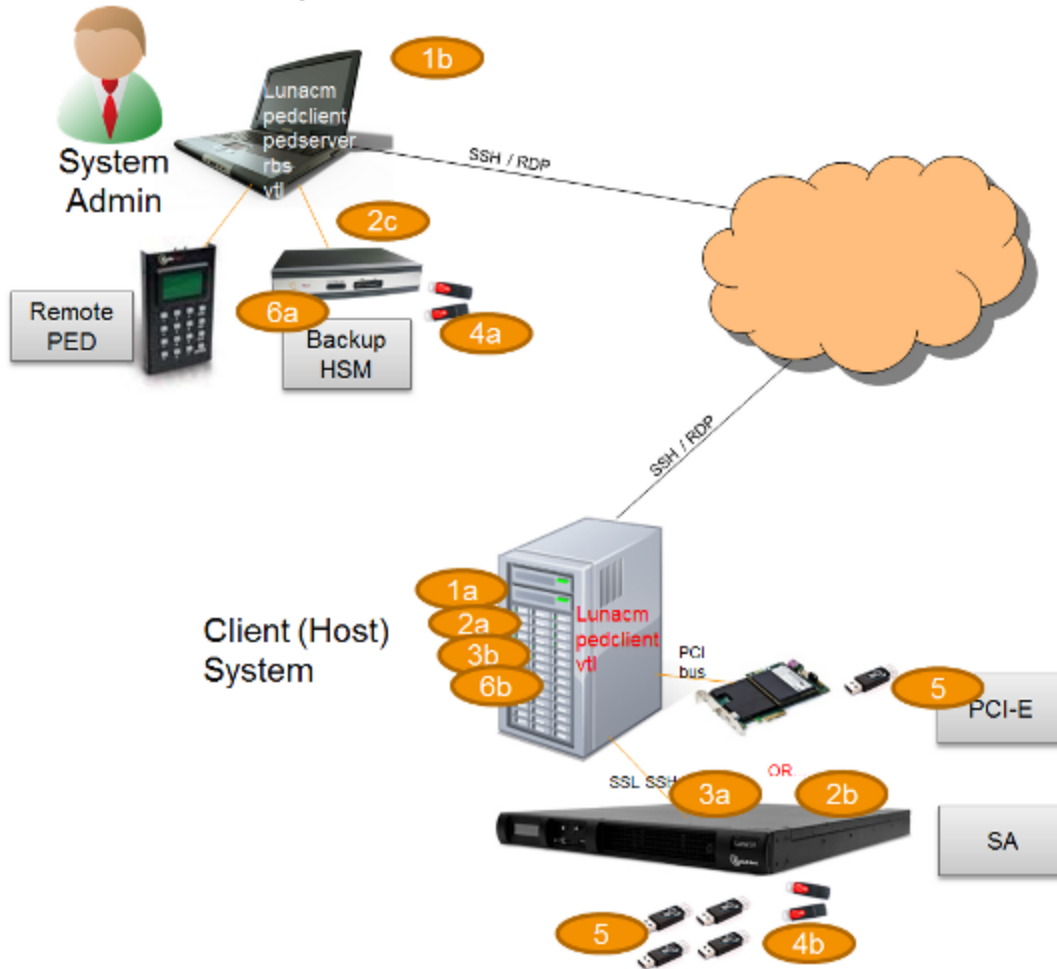
PedServer must reside (and run, waiting for calls) on any computer connected to a Remote PED.

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

Example

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

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



1	"Client (Host) System" (1a) is a client of the SafeNet Network HSM being backed up, but "System Admin" (1b) is not a client of SafeNet Network HSM.
2	LunaCM on "Client (Host) System" (2a) sees the primary (2b) and backup (2c) slots and controls the backup/restore.
3	Each SafeNet Network HSM (3a) partition is a slot visible to a "Client (Host) System" (3b) when Client (Host) System runs LunaCM.

4	Every slot on the backup (4a) must have same domain (red PED Key) as matching slot on the primary HSMs (4b).
5	Every primary HSM slot (partition) that is to be backed up or restored must be in login or activated state (black PED Keys (5)), so that the Client (Host) System can access it with LunaCM backup or restore commands.
6	Backup HSM (6a) is a slot visible to "Client (Host) System" (6b) when Client (Host) System runs LunaCM.

Backup HSM Installation, Storage, and Maintenance

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

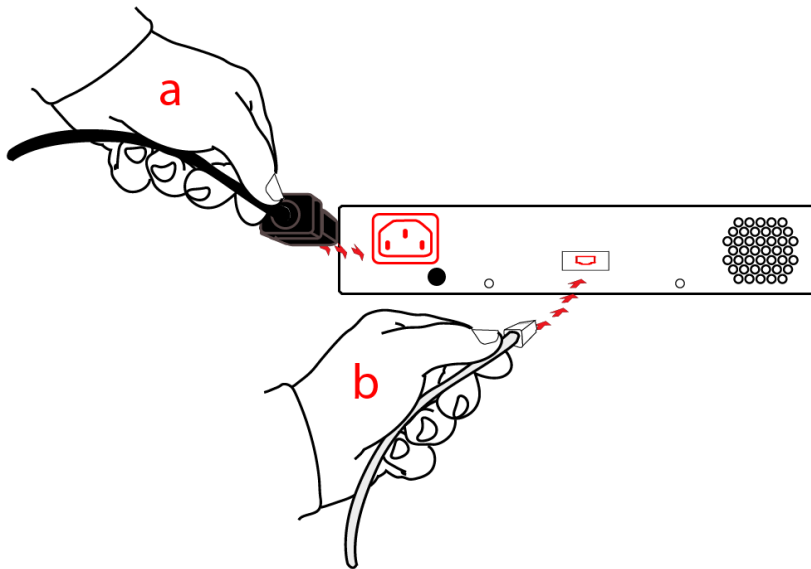
- "Connecting a Backup HSM" below
- "Disconnecting a Backup HSM " on page 54
- "Installing the Battery" on page 54
- "Backup HSM Storage and Maintenance" on page 57

Connecting a Backup HSM

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

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

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



PED-authenticated HSMs

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



External HSMs (Token-style and G5 style)

You can connect a SafeNet DOCK2 card reader for limited use with SafeNet Backup tokens (legacy G4 PCMCIA removable token-format HSMs). The removable-token backup HSM was used to backup legacy SafeNet Network 4.x HSMs and can be connected to SafeNet Network HSM 5.x or 6.x to restore the legacy key material as part of a one-way migration.

You can connect the more modern SafeNet USB HSM as an externally connected PKI slot, for use in the PKI Bundle option. Some customers use this arrangement to hold a root CA. The following caveats apply:

- The **token backup** commands can see and manage only the backup device, and not PKI devices.
- The **token pki** commands can see and manage only the PKI devices, and not backup devices.
- The PKI device must use PED authentication only, to be deployed.
- The **token pki update** commands update the capability and firmware for PKI devices.
- The process to move keys off G4 token HSMs (SafeNet CA4) is to migrate the keys to a K6 HSM (either the K6 inside SafeNet Network HSM, or the standalone K6 (SafeNet PCI-E HSM inside a host computer)) and then to SafeNet USB HSM. Cloning between G4 and G5 devices is not supported.



CAUTION: Migration is not supported to firmware 6.22.0. Migrate first to an HSM at a firmware version older than 6.22.0, and then update the HSM firmware to version 6.22.0 or newer.



CAUTION: Beginning with SafeNet HSM 6, we do not support PKI bundle using removable PCMCIA token HSMs (SafeNet CA4) and the SafeNet DOCK 2 reader. The SafeNet DOCK 2 reader is supported only for migration. If you need the PKI bundle function from removable tokens, do not upgrade.



Note: PPSO is not supported for the PKI-bundle configuration using SafeNet USB HSM. There is no provision to apply PPSO capability via SafeNet Network HSM to the externally connected SafeNet USB HSM. If the SafeNet USB HSM was removed to a host computer and updated to firmware 6.22.0 and had the PPSO capability applied (destructive operation), then returned to the SafeNet Network HSM to resume PKI-bundle operation, the interface has no provision to create a PPSO partition in the external HSM. Rather, a legacy-style partition would be created for PKI-bundle operation.

Disconnecting a Backup HSM

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

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

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

Installing the Battery

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

1



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

2



The battery compartment is to the right as you face the unit. The compartment cover is circular and has both raised dots and a recessed slot. Use finger-pressure against the dots, or the edge of a coin in the slot, to twist the battery compartment cover $\frac{1}{4}$ turn in a counter-clockwise direction. The cover should fall out easily.

3



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

4



Insert the battery, negative end first. The positive end (+) should protrude. The compartment is spring-loaded.

5



Use the battery compartment cover to push the battery into the compartment, against the spring tension. Maintaining the pressure, align the two tabs on the inside of the cover with the two recessed indentations at the top and bottom of the compartment opening. With a little jiggling and a few trial pushes, the tabs should settle into those recesses, allowing the cover to seat flush with the front of the SafeNet Remote Backup HSM. Maintain the inward pressure and twist the cover ¼ turn clockwise to lock it in place. The battery is installed.

6

Replace the front-panel cover by aligning the clips with their respective posts and pushing until the clips grab the posts and the cover snaps in place.

Backup HSM Storage and Maintenance

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

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

It is generally good practice to remove batteries when storing electronic devices, to preclude accidental damage from battery leakage. We use high-quality, industrial-grade batteries, that are unlikely to fail in a damaging fashion, but prudence suggests removing them, regardless. Also, if the unit is not in use, there is no need to maintain power to the RTC and NVRAM, so an externally stored battery will last longer (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 session objects (including working copies of stored keys) that are in use at the time. If the "originals" of those same objects are stored as HSM/partition objects, then they reside in non-volatile memory, and those are preserved.

There is no corruption of stored objects.

Where can I get a spare/replacement battery?

From any supplier that can match the specifications.

Technical Specifications:

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

Fast voltage recovery after long term storage and/or usage

Low self discharge rate

10 years shelf life

Operating temperature range -55 °C to +85 °C

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 °C

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.

Local Application-Partition Backup and Restore Using the Backup HSM

This section describes how to perform local backup and restore operations using the SafeNet Remote Backup HSM (Backup HSM). A local backup is defined as one in which the Backup HSM is local to the HSM or to the SafeNet HSM client workstation used to administer the HSM. To perform a local backup/restore, you can connect the Backup HSM using one of the following methods:

- to a USB port on the SafeNet Network HSM appliance. This method allows you use LunaSH to backup all of the SafeNet Network HSM partitions on the appliance that are owned by you, the HSM SO. It does not allow you to backup partitions that have their own SO. See ["Partition Backup and Restore Using a Backup HSM Connected Directly to a SafeNet Network HSM Appliance"](#) on the next page for details.

- to a USB port on the SafeNet HSM client workstation. This method allows you use LunaCM to backup any SafeNet Network HSM or SafeNet PCI-E HSM partitions that are visible as slots. See ["Partition Backup and Restore Using a Backup HSM Connected to a Local Client Workstation"](#) on page 63 for details.

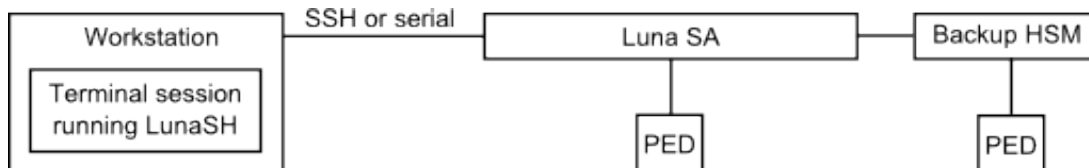
The backup operation can go from a source partition (on a SafeNet Network HSM) to an existing partition on the Backup HSM, or if one does not exist, a new partition can be created during the backup. The restore operation, however, cannot create a target partition on a SafeNet Network HSM; it must already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

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

Partition Backup and Restore Using a Backup HSM Connected Directly to a SafeNet Network HSM Appliance

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

Figure 1: Configuration for SafeNet Network HSM partition backup/restore using a Backup HSM connected directly to the SafeNet Network HSM appliance



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

The PEDs are required only if the SafeNet Network HSM is PED-authenticated. The appropriate SO (blue), partition (black) and domain (red) PED keys are required. The SO (blue), partition (black) PED keys can be the same for both devices, or can be different. Both devices must share the same domain (red) PED key value. Although two PEDs are recommended (one connected to the SafeNet Network HSM and one connected to the Backup HSM) you can use a single PED, if desired. If using a single PED, note that you can connect the PED to only one HSM at a time. You will need to disconnect it from the source (SafeNet Network HSM) HSM and connect to the target (SafeNet Remote Backup HSM) when PED operations are needed at those HSMs respectively.



Note: You can use this method to backup the partitions on the SafeNet Network HSM appliance that are owned by you, the HSM SO. You cannot use this method to backup partitions configured with their own SO. To backup a partition with SO, you must use LunaCM, as described in ["Partition Backup and Restore Using a Backup HSM Connected to a Local Client Workstation"](#) on page 63.

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

- Connect all the required components and open a terminal session to the SafeNet Network HSM appliance. See the following topics for details:
 - ["Open a Connection"](#) on page 1 in the *Configuration Guide*
 - ["Backup HSM Installation, Storage, and Maintenance"](#) on page 52

As soon as the PED is connected to a powered HSM it starts up and defaults to Local mode with the **Awaiting command...** prompt.

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

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

```
SafeNet Network HSM 6.0.0-25 Command Line Shell - Copyright (c) 2001-2014 SafeNet, Inc. All
rights reserved.
[myluna] lunash:>
```

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

```
[myluna] lunash:>token backup list
```

```
Token Details:
=====
Token Label:      BackupHSM
Slot:             6
Serial #:         7000179
Firmware:         6.22.0
Hardware Model:   SafeNet USB HSM
```

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

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

```
Partition Information:
```

```
=====
Partitions licensed on token: 20
Partitions created on token: 0
-----
```

There are no partitions.

```
Token Storage Information:
```

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

```
License Information:
```

```
=====
621010355-000      621-010355-000 G5 Backup Device Base
```



```

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

```

```

Command result : 0 (Success)
lunash:>

```

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

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

```

Type 'proceed' to continue the backup, or 'quit'
to abort this operation.

```

```

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

```

```

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

```

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

```

Luna PED operation required to login to token - use token Security Officer (blue) PED key.
Luna PED operation required to create a partition - use User or Partition Owner (black) PED
key.
Luna PED operation required to login to user on token - use User or Partition Owner (black)
PED key.
Luna PED operation required to generate cloning domain on the partition - use Domain (red)
PED key.

```

```

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

```

```

Command Result : 0 (Success)
[myluna] lunash:>

```

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

```

lunash:> token backup show -serial 667788
Token Details:
=====
Token Label:          BackupHSM
Serial #:             700179
Firmware:             6.22.0
Hardware Model:       SafeNet USB HSM
Authentication Method: PED keys
Token Admin login status: Logged In
Token Admin login attempts left: 3 before Token zeroization!

```

```

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

Token Storage Information:
=====

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

License Information:
=====

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

Command result : 0 (Success)
lunash:>

```

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

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

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

- ["Open a Connection" on page 1](#) in the *Configuration Guide*
- ["Backup HSM Installation, Storage, and Maintenance" on page 52](#)

As soon as the PED is connected to a powered HSM it starts up and defaults to Local mode with the **Awaiting command...** prompt.

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

```

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

SafeNet Network HSM 5.1.0-25 Command Line Shell - Copyright (c) 2001-2011 SafeNet, Inc. All
rights reserved.
[myluna] lunash:>

```

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

```

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

```

```

CAUTION: Are you sure you wish to erase all objects in the
          partition named:                p1
          Type 'proceed' to continue, or 'quit' to quit now.

```

```

> proceed
Warning: You will need to attach Luna PED to the SafeNet Backup HSM to complete this operation.

        You may use the same Luna PED that you used for SafeNet Network HSM.

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

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

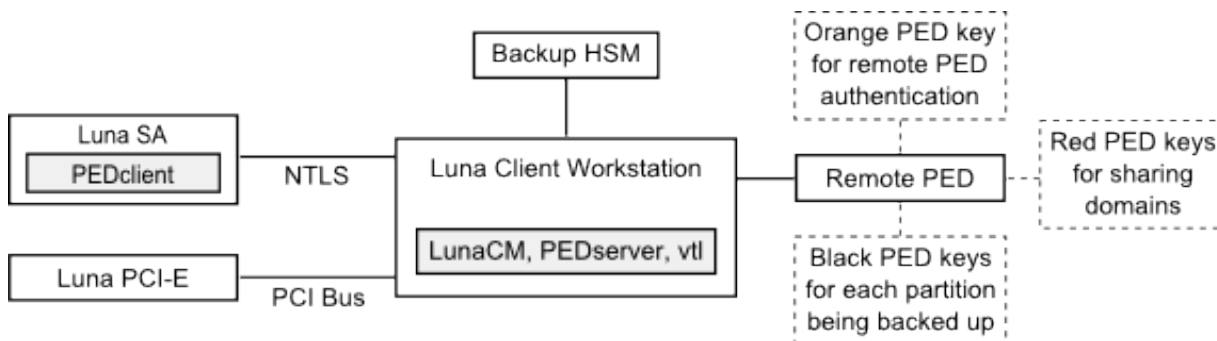
Command Result : 0 (Success)
[myluna] lunash:>

```

Partition Backup and Restore Using a Backup HSM Connected to a Local Client Workstation

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

Figure 2: Configuration for SafeNet Network HSM/PCI-E partition backup/restore using a Backup HSM connected to a local client workstation



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

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

If you are backing up PED-authenticated partitions, you require a PED. If you want to backup SafeNet Network HSM partitions, the PED must have remote capability (Remote PED). Remote PED uses the pedserver/pedclient processes running on the SafeNet HSM client workstation and on the SafeNet Network HSM appliance to provide remote PED services for the network-attached SafeNet Network HSM appliance. The PED provides authentication for all of the

attached HSMs (the USB-connected SafeNet Remote Backup HSM, the NTLS-connected SafeNet Network HSM, and the PCI bus-connected SafeNet PCI-E HSM). Every slot on the backup must have same domain (red PED Key) as the matching slot on the source HSMs.



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

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

1. Configure the remote PED, as described in "Remote PED" on page 370.
2. Start the LunaCM utility on the SafeNet HSM client workstation.

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

```
LunaCM V2.3.3 - Copyright (c) 2006-2014 SafeNet, Inc.
```

```
Available HSM's:
```

```
Slot Id -> 1
HSM Label -> SA52_P1
HSM Serial Number -> 500409014
HSM Model -> LunaSA
HSM Firmware Version -> 6.22.0
HSM Configuration -> SafeNet Network HSM Slot (PED) Signing With Cloning Mode
HSM Status -> OK
```

```
Slot Id -> 2
HSM Label -> BackupHSM Serial Number -> 700101
HSM Model -> G5Backup
HSM Firmware Version -> 6.22.0
HSM Configuration -> Remote Backup HSM (PED) Backup Device
HSM Status -> OK
```

```
Current Slot Id: 1
```

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

```
lunacm:> slot set slot 1
```

```
Current Slot Id: 1 (SafeNet Network HSM Slot 6.10.1 (PED) Signing With Cloning Mode)
```

```
Command Result : No Error
```

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

```
lunacm:>ped get
```

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

```
Command Result : No Error
```

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

```
Command Result : No Error
```

```
lunacm:> ped get
```

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

```
Command Result : No Error
```

```
lunacm:>
```

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

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

```
lunacm:> par login
```

```
Option -password was not supplied. It is required.
```

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

```
User is activated, PED is not required.
```

```
Command Result : No Error
```

```
lunacm:>
```

6. Disconnect the logical PED connection from your source HSM (slot 1 in this example), and connect to the Backup HSM (slot 2 in this example). The PED remains physically connected by USB cable to the SafeNet HSM client workstation, and remains in Remote mode - you are merely changing slots that are in conversation with that PED.

- a. First, tell the SafeNet Network HSM to disconnect from Remote PED.

```
lunacm:> ped disconnect
```

```
Are you sure you wish to disconnect the remote ped?
```

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

```
Command Result : No Error
```

- b. Then tell the Backup HSM to connect to Remote PED (it makes no difference that the PED and the Remote Backup HSM are USB-connected to the same workstation/laptop; when use of "Remote PED" is invoked by command "ped connect" and verified by "ped get", all HSM-PED interaction takes place between "pedclient" running on that workstation and "pedserver", also running on that workstation).

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

```
Command Result : No Error
```

```
lunacm:> ped get -slot 2
```

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

Command Result : No Error

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

```
lunacm:> partition archive backup -slot 2 -par SAbck1
```

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

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

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

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

```
Verifying that all objects can be backed up...
85 objects will be backed up.
```

```
Backing up objects...
Cloned object 99 to partition SAbck1 (new handle 19).
Cloned object 33 to partition SAbck1 (new handle 20).
Cloned object 108 to partition SAbck1 (new handle 23).
Cloned object 134 to partition SAbck1 (new handle 24).
Cloned object 83 to partition SAbck1 (new handle 25).
Cloned object 117 to partition SAbck1 (new handle 26).
Cloned object 126 to partition SAbck1 (new handle 27).
Cloned object 65 to partition SAbck1 (new handle 28).
Cloned object 140 to partition SAbck1 (new handle 29).
Cloned object 131 to partition SAbck1 (new handle 30).
Cloned object 94 to partition SAbck1 (new handle 31).
Cloned object 109 to partition SAbck1 (new handle 35).
Cloned object 66 to partition SAbck1 (new handle 36).
Cloned object 123 to partition SAbck1 (new handle 39).
Cloned object 74 to partition SAbck1 (new handle 40).
Cloned object 50 to partition SAbck1 (new handle 44).
Cloned object 43 to partition SAbck1 (new handle 45).
Cloned object 52 to partition SAbck1 (new handle 46).
Cloned object 124 to partition SAbck1 (new handle 47).
Cloned object 115 to partition SAbck1 (new handle 48).
```

```
Backup Complete.
```

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

Command Result : No Error

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

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

1. Create a target partition for the restore operation on the HSM you are restoring to, if it does not already exist, and

register the partition with the SafeNet HSM client workstation so that it is visible as a slot in LunaCM.

2. Start the LunaCM utility on the SafeNet HSM client workstation.

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

```
LunaCM V2.3.3 - Copyright (c) 2006-2014 SafeNet, Inc.
```

```
Available HSM's:
```

```
Slot Id -> 1
HSM Label -> SA52_P1
HSM Serial Number -> 500409014
HSM Model -> LunaSA
HSM Firmware Version -> 6.22.0
HSM Configuration -> SafeNet Network HSM Slot (PED) Signing With Cloning Mode
HSM Status -> OK
```

```
Slot Id -> 2
HSM Label -> BackupHSM Serial Number -> 700101
HSM Model -> G5Backup
HSM Firmware Version -> 6.22.0
HSM Configuration -> Remote Backup HSM (PED) Backup Device
HSM Status -> OK
```

```
Current Slot Id: 1
```

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

```
lunacm:> slot set slot 1
```

```
Current Slot Id: 1 (SafeNet Network HSM Slot 6.22.0 (PED) Signing With Cloning Mode)
```

```
Command Result : No Error
```

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

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

```
Command Result : No Error
```

```
lunacm:> ped get
```

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

```
Command Result : No Error
```

```
lunacm:>
```

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

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

```
lunacm:> par login
```

```
Option -password was not supplied. It is required.
```

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

```
User is activated, PED is not required.
```

```
Command Result : No Error
```

```
lunacm:>
```

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

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



Note: In the command above, you could have used **-add** instead of **-replace**. Adding might result in unwanted behaviors, such as having two keys with the same label, if one existed in the HSM Partition and one on the backup token. The two would be assigned different handles, however.

Remote Application-Partition Backup and Restore Using the Backup HSM

This section describes how to perform remote backup and restore operations using the SafeNet Remote Backup HSM (Backup HSM). It contains the following sections:

- "Overview" below
- "Configuring the Remote Backup Service (RBS)" on page 71
- "Backing Up an Application Partition to a Remotely Located Backup HSM" on page 73
- "Restoring an HSM Partition From a Remotely Located Backup HSM" on page 78

Overview

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

To use RBS, you do the following:

- configure it to define which of the Backup HSMs connected to the workstation running RBS that you want to make available to other SafeNet HSM client workstations or SafeNet Network HSM appliances for performing remote backups.
- register the workstation running RBS with any SafeNet HSM client workstations or SafeNet Network HSM appliances that you want to be able to use the remote Backup HSMs.
- start the RBS service/daemon.

Once RBS is configured and running, the SafeNet HSM client workstations or SafeNet Network HSM appliances registered with the workstation running RBS can see its available Backup HSMs as slots in LunaCM (SafeNet HSM client workstation) or LunaSH (SafeNet Network HSM appliance). To perform backup and restore operations using the remote Backup HSMs, you open a LunaCM or LunaSH session, as relevant, on the SafeNet HSM client workstation or

SafeNet Network HSM appliance used to host the slot you want to backup, and specify the slot for the remote Backup HSM as the slot to use for the backup/restore operation.

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

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



Note: Remote PED (PEDServer) is supported on Windows only.

Configurations for Remote Backup of a SafeNet Client Workstation Slot

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

Figure 1: Configuration for remote backup of a SafeNet HSM client workstation slot with the remote PED connected to the backup workstation

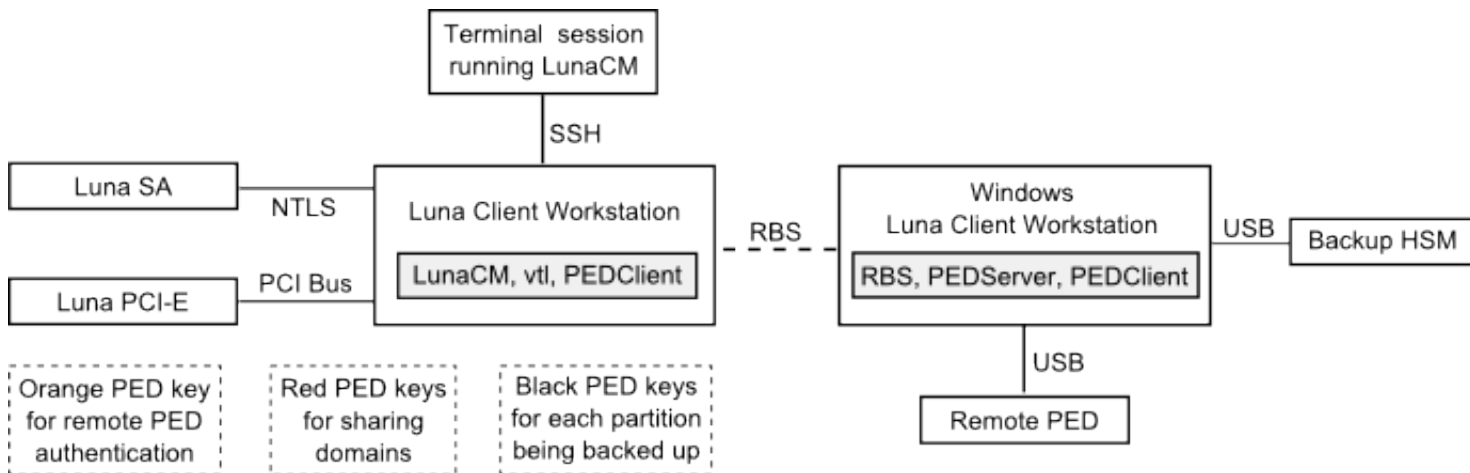
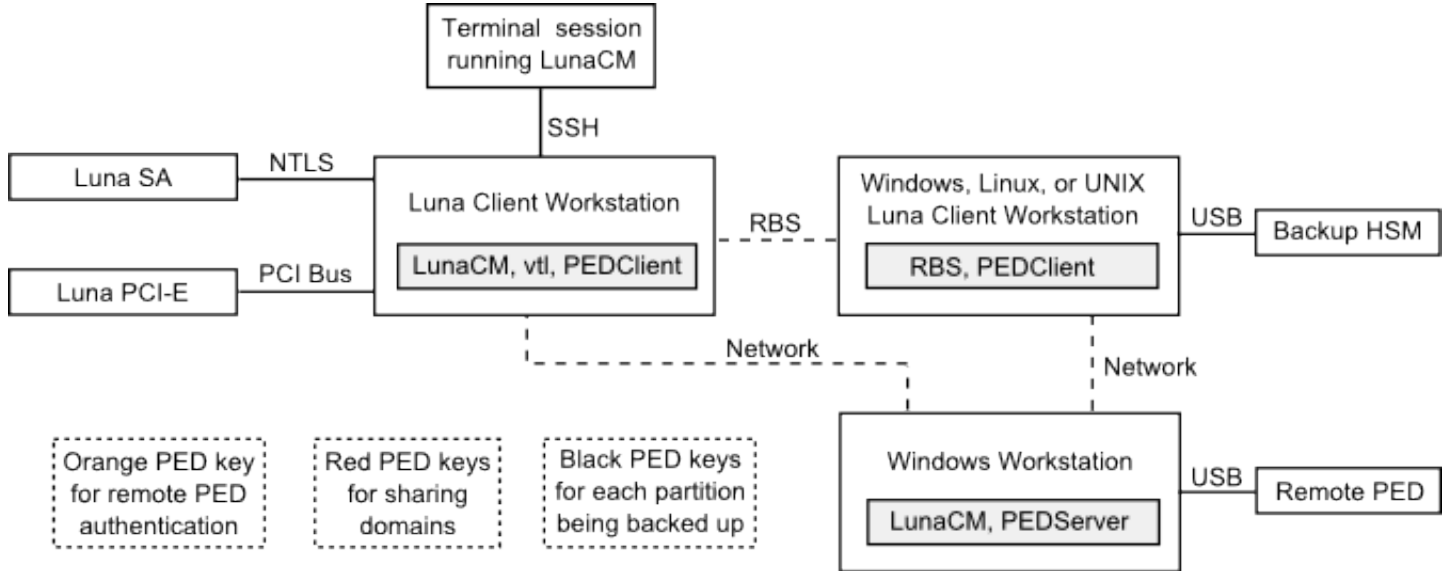


Figure 2: Configuration for remote backup of a SafeNet HSM client workstation slot with the remote PED connected to a separate workstation



Configurations for Remote Backup of a SafeNet Network HSM Appliance

The possible configurations for performing a remote backup of a SafeNet Network HSM appliance are illustrated in the following figures. Only PED-authenticated backup configurations are shown.

Figure 3: Configuration for remote backup of a SafeNet Network HSM appliance with the remote PED connected to the backup workstation

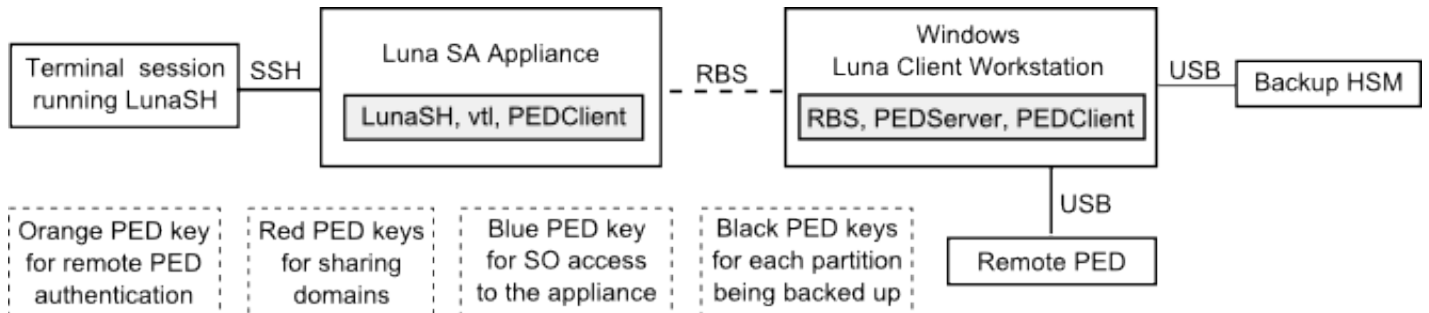
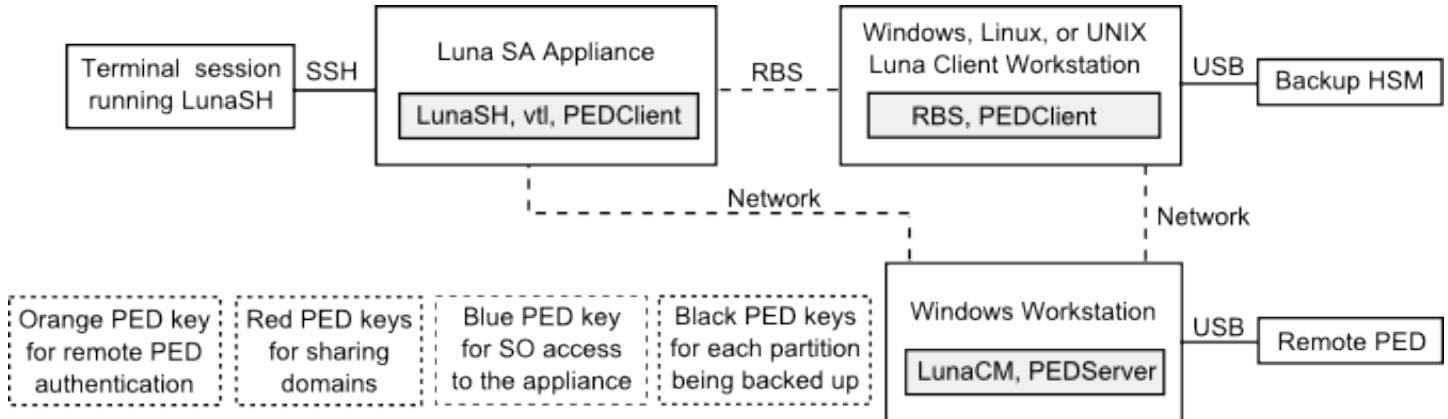


Figure 4: Configuration for remote backup of a SafeNet Network HSM appliance with the remote PED connected to a separate workstation



Configuring the Remote Backup Service (RBS)

RBS is not a standalone feature. It is a service that facilitates certain scenarios when backing-up HSM partitions or restoring onto those partitions, using a backup HSM that is distant from the primary HSM and its host or client. RBS is run on the computer that hosts the SafeNet Remote Backup HSM, only. RBS is a separate option at software installation time. You do not need it on all client/admin computers, but it doesn't hurt to have it installed. Running RBS also requires running PEDClient on that computer, as well as on the distant primary - the paired instances of PEDClient form the communications link that makes RBS possible.

RBS requires PEDClient on both the RBS client and RBS server ends.

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

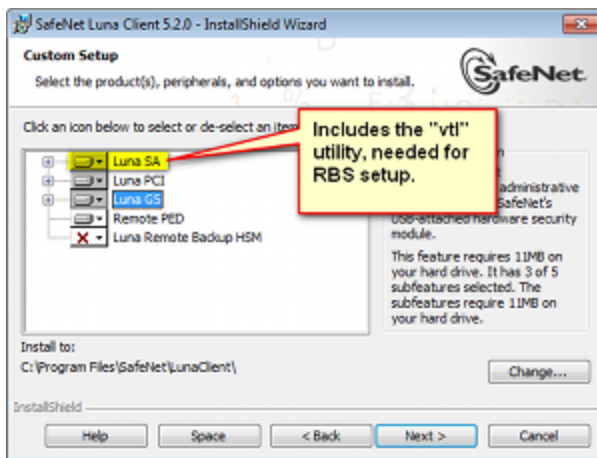
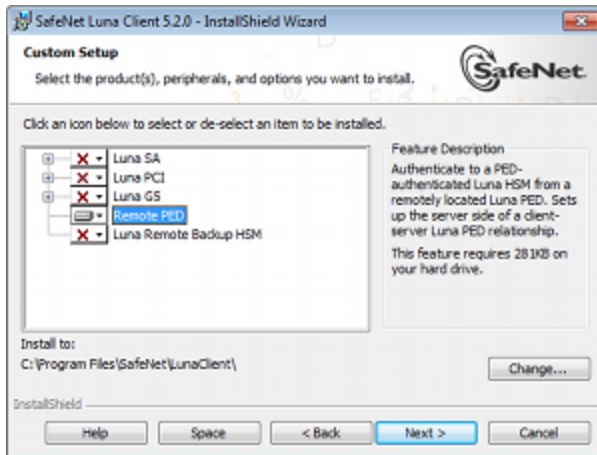
However, PEDClient is also used in the communication component of Remote Backup Service. So, PEDClient should run on all the platforms that have HSMs - where a SafeNet USB HSM or SafeNet PCI-E HSM is installed (PEDClient is already inside SafeNet Network HSM 5.2 and newer...) - and also on any system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

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

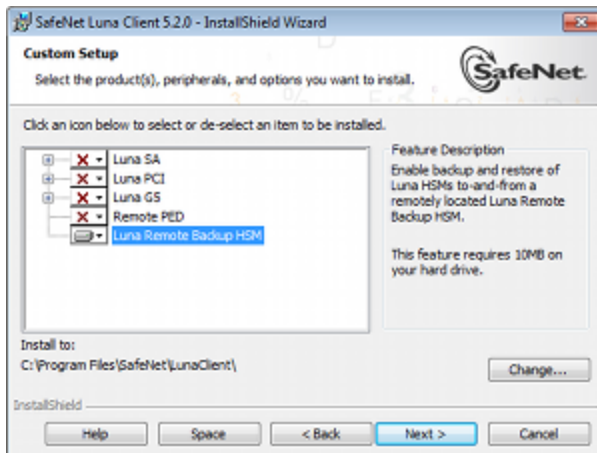
If you consolidate your HSM administration (including Remote PED) on the same computer with your SafeNet Remote Backup HSM, you would have both PEDClient and PEDServer installed there. We observe that a majority of customers combine administrative functions this way, on a laptop or a workstation that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

To configure RBS

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



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



3. Run **rbs --genkey** to generate the **server.pem** to establish the Remote Backup Service between the Backup host and the host/client for the primary HSM. The location of the **server.pem** file can be found in the **Chrystoki.conf** /**crystoki.ini** file.
4. Run **rbs --config** to specify the devices to support.

5. Run **rbs --daemon** to launch the rbs daemon (Linux and UNIX) or the rbs console application (on Windows, it closes after every use) .
6. Create the client certificate (if not already done) :
vtl createCert -n <host_ip_address>
7. Use **scp** (Unix/Linux) or **pscp** (Windows) to copy the certificate generated earlier (**server.pem**) to your primary HSM host computer (or SafeNet Network HSM appliance).

```
# scp root@172.20.9.253:/usr/safenet/lunaclient/rbs/server/server.pem .
root@172.20.9.253's password: *****
server.pem | 1 kB | 1.2 kB/s | ETA: 00:00:00 | 100%
```
8. Run **vtl** on the host computer (or appliance) to add the RBS server to the server list.

```
vtl add -n 172.20.9.253 -c server.pem
New server 192.20.9.253 successfully added to server list.
vtl list
Server: 192.20.9.82 HTL required: no
Server: 192.20.9.253 HTL required: no
```



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

Backing Up an Application Partition to a Remotely Located Backup HSM

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

Prerequisites

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

Quantity	Description
1	SafeNet HSM 5.2 or newer
1	Windows computer with SafeNet Network HSM 5.2 (or newer) client software installed
1	SafeNet Remote Backup HSM
1	Set of PED Keys imprinted for the source HSM and partitions
1	SafeNet PED 2 (Remote PED with f/w 2.4.0 or later)*
1	Power cable for SafeNet PED 2 (Remote)
2	USB to mini USB cable for SafeNet PED 2 (Remote) and SafeNet Remote Backup HSM



Note: The SafeNet PED that is connected to the Windows computer, in order to perform Remote PED operations with the distant SafeNet Network HSM appliance, must be a SafeNet PED 2 (remote-capable version) and is used in Remote mode and in local mode. You also have

the option to connect a second SafeNet PED, which can be Remote capable or can be a local-only version, to the SafeNet Backup HSM. This allows you to leave the Remote capable SafeNet PED connected to the workstation in Remote mode.

Assumptions

The following examples assume that you have set up RBS, as described in "Configuring the Remote Backup Service (RBS)" on page 71, and have prepared for the backup, as follows:

- the Backup HSM and the HSMs/partitions you want to back up are initialized with appropriate keys (blue SO and black Partition Owner/User PED Keys, which can be the same for both devices, or can be different)
- Both devices **must** share the **same** domain or RED key value.
- The workstation (Windows computer) has Remote PED and SafeNet Remote Backup software package installed including the appropriate driver, if you are using it to
- For SafeNet Network HSM, NTLS is established between your workstation computer, acting as a SafeNet Network HSM client, and the distant SafeNet Network HSM - that is, the workstation is registered as a client with the partition.
- A Remote PED session key (orange RPV key) has been created and associated with the distant SafeNet HSM.

To Backup an Application Partition to a Remotely Located Backup HSM

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

Set up the remote PED

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

From	Using	To
Workstation	USB	Remote PED (SafeNet PED IIr in Remote mode)
DC power receptacle on Remote PED	PED Power Supply	Mains AC power (wall socket)
Workstation	USB	SafeNet Remote Backup HSM
SafeNet Remote Backup HSM	Power Cord	Mains AC power (wall socket)

- At the Remote SafeNet PED (SafeNet PED 2 with remote capability, connected to the USB port of the workstation), do the following:
 - press **<** on the PED keypad to exit local mode.
 - press **7** to enter remote mode.
- Run **PEDServer** to start the remote PED service on the administrative workstation (Windows) computer, as follows:

- In a Command Prompt (DOS) window, change directory to the location of the **PEDServer.exe** file and run that file:

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

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

Example (substitute the actual IP address of your workstation computer)--

```
lunash:> hsm ped connect -ip 192.2.12.16 -port 1503
```

or

```
lunacm:> hsm ped connect -ip 192.2.12.16 -port 1503
```

Insert the orange RPV PED Key that matches the RPV of the distant SafeNet HSM.

The Remote PED Client in the SafeNet Network HSM appliance or in the SafeNet HSM client workstation establishes a connection with the listening PEDserver on your remote PED workstation.

Backup a slot to the remotely located backup HSM



Note: The following steps apply to LunaCM only. For LunaSH, follow the procedure "To backup a SafeNet Network HSM partition to a directly connected Backup HSM" on page 59. Use the **token backup list** and **token backup show** commands to ensure that the remote Backup HSM is visible.

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

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

```
LunaCM V6.0.0 - Copyright (c) 2006-2015 SafeNet, Inc.
```

```
Available HSM's:
```

```
Slot Id -> 1
HSM Label -> SA82_P1
HSM Serial Number -> 16298193222733
HSM Model -> LunaSA
HSM Firmware Version -> 6.22.0
HSM Configuration -> Luna User Partition, With SO (PED) Signing With Cloning Mode
HSM Status -> OK
```

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

```
Slot Id -> 3
HSM Label -> G5backup
HSM Serial Number -> 700101
HSM Model -> G5Backup
```

```
HSM Firmware Version -> 6.10.1
HSM Configuration -> Remote Backup HSM (PED) Backup Device
HSM Status -> OK
```

```
Current Slot Id: 1
```

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

```
lunacm:> slot set slot 1
```

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

```
Command Result : No Error
```

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



Note: The PEDServer must already have been set up at that host.

```
lunacm:>ped get
```

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

```
Command Result : No Error
```

```
lunacm:> ped connect ip 172.20.10.190
```

```
Command Result : No Error
```

```
lunacm:> ped get
```

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

```
Command Result : No Error
```

9. Skip this step if your source partition is activated.

Log into the partition (this takes place at the currently selected slot). This step is needed only if the partition you are about to backup is not already in the activated state.

Example for HSM with firmware 6.22.0 or newer:

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

```
Option -password was not supplied. It is required.
```

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

```
User is activated, PED is not required.
```

```
Command Result : No Error
```

Example for HSM with firmware older than version 6.22.0:

```
lunacm:> par login
```


Option -password was not supplied. It is required.

Enter the password: *****

User is activated, PED is not required.

Command Result : No Error

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

```
lunacm:> ped disconnect
```

Are you sure you wish to disconnect the remote ped?

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

Command Result : No Error

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

Command Result : No Error

```
lunacm:> ped get -slot 3
```

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

Command Result : No Error

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

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

Logging in as the SO on slot 3.

Please attend to the PED.

Creating partition SAbck1 on slot 3.

Please attend to the PED.

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

Please attend to the PED.

Creating Domain for the partition SAbck1 on slot 3.

Please attend to the PED.

Verifying that all objects can be backed up...

85 objects will be backed up.

Backing up objects...

Cloned object 99 to partition SAbck1 (new handle 19).

Cloned object 33 to partition SAbck1 (new handle 20).

Cloned object 108 to partition SAbck1 (new handle 23).

.

.

```
.
Cloned object 78 to partition SAbck1 (new handle 128).
Cloned object 88 to partition SAbck1 (new handle 129).
Cloned object 40 to partition SAbck1 (new handle 130).
```

```
Backup Complete.
```

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

```
Command Result : No Error
```

12. The backup operation is complete.

Restoring an HSM Partition From a Remotely Located Backup HSM

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

To restore an application partition from a remotely located backup HSM

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



Note: The following steps apply to LunaCM only. For LunaSH, follow the procedure "To restore a SafeNet Network HSM partition from a directly connected Backup HSM" on page 62. Use the **token backup list** and **token backup show** commands to ensure that the remote Backup HSM is visible.

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

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

```
LunaCM v6.0.0 - Copyright (c) 2006-2015 SafeNet, Inc.
```

```
Available HSMs:
```

```
Slot Id ->          0
Label ->
Serial Number ->    16298193222733
Model ->            LunaSA
Firmware Version -> 6.22.0
Configuration ->    Luna User Partition With SO (PED) Signing With Cloning Mode
Slot Description -> Net Token Slot

Slot Id ->          1
Label ->
Serial Number ->    16298193222735
Model ->            LunaSA
Firmware Version -> 6.22.0
Configuration ->    Luna User Partition With SO (PED) Signing With Cloning Mode
```

```

Slot Description ->      Net Token Slot

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

Slot Id ->              3
Label ->               SAbck1
Serial Number ->       700101
Model ->               G5Backup
Firmware Version ->   6.10.4
Configuration ->      Luna User Partition With SO (PED) Signing With Cloning Mode
Slot Description ->   User Token Slot

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

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

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

Current Slot Id: 0

```

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

```
lunacm:>ped get
```

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

```
Command Result : No Error
```

3. Connect to Remote PED.

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

Command Result : No Error

(Causes the currently selected slot in lunacm (still slot 0 in this example) to connect to the remote PED.)

4. Log into the partition to which you want to restore.



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

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

```
enter password: *****
```

```
Please attend to the PED.
```

Command Result : No Error

```
lunacm:> ped disconnect
```

```
Are you sure you wish to disconnect the remote ped?
```

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

Command Result : No Error

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



Note: In this example, the SafeNet Network HSM partition, to which we will restore objects, is visible in lunacm at slot 0 because it is linked to this SafeNet HSM client by NTLS, while this Client is registered to that partition at the SafeNet Network HSM.

The SafeNet Remote Backup HSM is visible in lunacm, at slot 3 in this case, because it is linked by the RBS connection that you previously established (see "To Configure RBS" above in this chapter); that is, pedclient is running on this Client, and pedclient and rbs.exe are running on the Backup HSM's host, with each other identified as their partner in the RBS link.

5. Connect the Remote PED to the Backup HSM (which, in this example, is slot 3).

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

Command Result : No Error

```
lunacm:> ped get
```

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

Command Result : No Error

```
lunacm:> ped get slot 3
```

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

Command Result : No Error

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

6. Restore to the current slot (slot 0) from the slot that corresponds to the Backup HSM (slot 3).

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

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

Please attend to the PED.

Verifying that all objects can be restored...

85 objects will be restored.

Restoring objects...

Cloned object 19 from partition SAbck1 (new handle 20).

Cloned object 20 from partition SAbck1 (new handle 21).

Cloned object 23 from partition SAbck1 (new handle 22).

.

.

.

Cloned object 128 from partition SAbck1 (new handle 137).

Cloned object 129 from partition SAbck1 (new handle 138).

Cloned object 130 from partition SAbck1 (new handle 139).

Restore Complete.

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

Command Result : No Error

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

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

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

HSM Storage Information for slot 3:

Total HSM Storage Space:	16252928
Used HSM Storage Space:	43616
Free HSM Storage Space:	16209312

```
Number Of Allowed Partitions: 20
Number Of Allowed Partitions: 1
```

```
Partition list for slot 3
```

```
Number of partition: 1
```

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

```
Command Result : No Error
```

```
lunacm:>
```

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

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

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

Small Form Factor Backup

The small form factor (SFF) backup feature is available for PED-authenticated SafeNet HSMs only.



Note: A SafeNet PED is required for SFF backup. A SafeNet PED with Remote capability is recommended for SFF backups. See the Customer Release Notes for more information.

Characteristics

Small form factor backup is mediated by SafeNet PED and uses SafeNet eToken 7300 USB devices as the repository for archived cryptographic objects.



- The eToken 7300 is Common Criteria validated and tamper-evident.
- SFF backup is supported for SafeNet Network HSM and SafeNet PCI-E HSM. Small Form-Factor Backup is not supported for SafeNet USB HSM (no capability update file is available). To back up a SafeNet USB HSM, clone the contents to another SafeNet USB HSM.)
- One eToken 7300 can back up one HSM partition.
- Backup and restore can be performed to or from an eToken 7300 inserted into a locally-connected or remotely-connected SafeNet PED (via PedServer).
- A capability update file (CUF) must be purchased and applied to each HSM (serial number specific) that is to use the SFF Backup feature.



Note: Using SFF backup imposes some constraints, and affects other features like HA. See "Cloning and SFF Backup Option Use Cases" on page 86 for more detailed information.



Required Elements

SFF backup requires:

- SafeNet Software version 5.4.0 or newer
- HSM firmware version 6.21.0 or newer
- A SafeNet PED with firmware version 2.6.0-6 or newer. A remote PED is recommended.
- Source SafeNet HSM must be cloning type, only - not applicable to Key Export (KE) HSMs
- Backup to a remotely located SFF backup requires a remote PED - a local-only PED is not field-upgradable to remote capability
- HSM must have the SFF backup capability update applied (this is a purchased option)



CAUTION: The SFF backup capability update is a destructive change to your HSM, meaning that the upgrade enforces HSM initialization and all contents will be lost. Back up any important keys or objects before the upgrade. You can recreate your partition(s) and restore your objects after the HSM has been re-initialized, following the application of the SFF backup capability upgrade.

Configuration

Small Form-Factor Backup requires that the SafeNet configuration file **crystoki.ini** (Windows) or **Chrystoki.conf** (Linux/UNIX) must have two specific settings:

- CommandTimeoutPedSet = 720000
- PEDTimeout2 = 200000

Newly installed/created SafeNet HSM client configuration files have the necessary entries, with the correct values, but pre-existing clients might be missing an entry or might have an insufficient value assigned.

To configure Linux/UNIX clients

On Linux clients the **Chrystoki.conf** file is saved upon SafeNet HSM client un-installation, and re-used on later installation. Manually run the following commands if needed:

1. If **CommandTimeoutPedSet** is missing in the Luna section run this command to add it:

```
/usr/safenet/lunaclient/bin/configurator setValue -s Luna -e CommandTimeoutPedSet -v 720000
```

2. If **PEDTimeout2** value is smaller than 200000 run this command:

```
/usr/safenet/lunaclient/bin/configurator setValue -s Luna -e PEDTimeout2 -v 200000
```

To configure Windows clients

On Windows clients, already having **crystoki.ini**, any new entry provided by the newer release of the SafeNet HSM client is added to the file. But existing entry values are not modified. Manually edit the **crystoki.ini** file and modify the needed entries as follows:

1. Set **CommandTimeoutPedSet** to 720000
2. Set **PEDTimeout** to 200000.

To Switch Off Small Form-Factor Backup

If you have concerns about the physical security of your HSMs, and wish to ensure that sensitive application partition contents cannot be backed-up onto a very portable, concealable SFF token, then simply do not purchase or apply a Small Form-Factor capability update for that HSM.

If the SFF Capability Update has been installed, and for any reason you wish to disable the ability to backup HSM content, or application partition objects, to a Small Form-Factor device, you must disable HSM Policy 38.



WARNING! Disabling SFF is HSM-wide and is destructive, meaning that HSM contents and partitions are lost. Re-initialization is required, and lost objects must be re-created or must be restored from a SafeNet Backup HSM or by synchronization in an HA group.

To disable the Small Form-Factor feature

1. Enter the following command. You must be logged in as the HSM SO.

```
lunacm:>hsm changehsmpolicy -policy 38 -value 0
```

Using Small Form Factor Backup

SFF backup requires that you install the serialized¹ Capability Update File (CUF) to your SafeNet HSM at firmware 6.21.0 or higher.



Note: SFF is supported on PED-authenticated HSMs only.

¹The Capability Update (or other software) is associated with a specific serial number and is meant to be used only with the HSM that carries that corresponding serial number. That is, the installation or upgrade command verifies that serial numbers match, before proceeding. This is used with purchased upgrades, and is also sometimes applied to Alpha or Beta versions of patches or other software, to ensure that unfinished, not-formally-released versions do not escape our verification and final approval process.

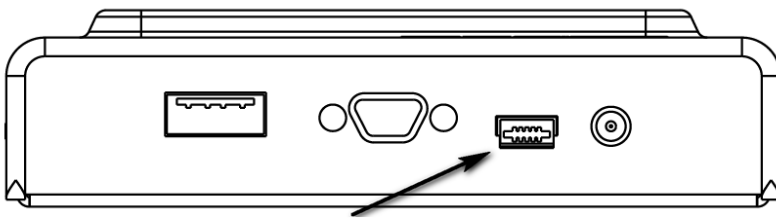
For SafeNet PCI-E HSM and SafeNet USB HSM, obtain the SFF backup capability upgrade from SafeNet, extract the update file and the authcode file on the HSM host computer, and run the lunacm **hsm updatecap** command. See "**hsm updatecap**" on page 1 in the *LunaCM Command Reference Guide* for details.

For SafeNet Network HSM, obtain the SFF backup capability upgrade from SafeNet, upload the spkg file to the appliance, install it with lunash **package update** command, and then use the **hsm update capability** command to apply the resulting CUF to the HSM. See "**package update**" on page 1 and "**hsm update capability**" on page 1 in the *LunaSH Command Reference Guide* for details.

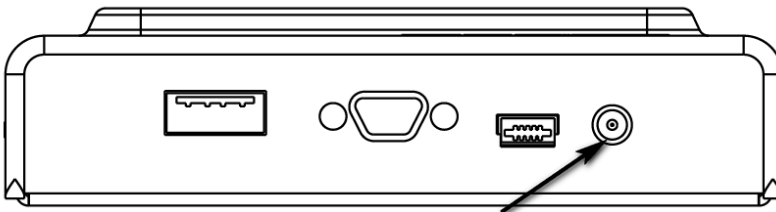
SFF backup requires a SafeNet PED as the interface for the eToken 7300.

Remote Connection

For remote SFF backup, connect the Remote PED to a USB port on the computer running the backup, and to the USB-mini port on the PED.



Also connect the PED power block to the PED and to an AC power outlet. The mini-USB connection is not sufficient or reliable to power the PED.



Preparing to Backup

1. Ensure that PEDTimeout2 is set to at least 200000 in the crystoki.ini file.
2. Run **pedserver** on the host system to which the PED is connected. You can optionally specify the current IP address of that computer, and the desired port, as illustrated below:

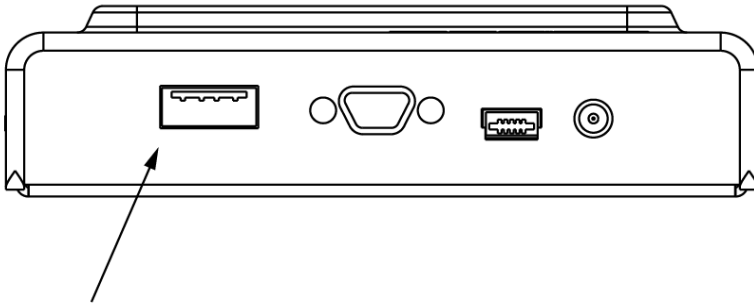
```
C:\Program Files\SafeNet\LunaClient>pedserver -mode start -ip 192.20.10.109 -port 1503
Ped Server Version 1.0.5 (10005)
Failed to load configuration file. Using default settings.
```

```
Starting background process
Background process started
Ped Server Process created, exiting this process.
C:\Program Files\SafeNet\LunaClient>
```

3. Start the PED client on the HSM host to provide the channel between the HSM and the PED where the eToken 7300 is to connect.
4. Use the LunaCM **ped connect** command to establish the connection, as illustrated below:

```
lunacm:> ped connect -ip 192.20.10.109 -port 1503
Command Result : No Error
```

5. Log in, as User, to the desired HSM partition that is to be the source of material for backing up.
6. Insert the SFF backup eToken into the PED Key connector on the top of the SafeNet PED, and wait for the lights to steady.



7. Run the backup command, specifying the eToken as the backup target, and any appropriate label:

```
partition archive backup -slot etoken -label mybackup
```

You can optionally specify a list (comma-delimited) of object handles, otherwise the default is that all partition objects are backed up.

Depending on the number of objects to back up, the operation can take many minutes.

8. When the operation completes with a success message at the command line, and the lights on the eToken have stopped flashing, remove the SFF backup eToken to safe storage.

Restoring

To restore from a SFF backup eToken, prepare the SafeNet PED as described above.

1. Log in, as User to the desired HSM partition that is to receive the restored material.
2. Insert the SFF backup eToken into the PED Key connector on the top of the SafeNet PED, and wait for the lights to steady.
3. Run the restore command, specifying the eToken as the backup target, and any appropriate label:

```
partition archive restore -slot etoken -label mybackup
```

Depending on the number of objects to restore, the operation can take many minutes.

4. When the operation completes with a success message at the command line, and the lights on the eToken have stopped flashing, remove the SFF backup eToken.
5. Verify the objects on the restored partition.

Cloning and SFF Backup Option Use Cases

This section describes the compatibility of small form factor (SFF) backup with HSM-to-HSM cloning in various configurations.



Note: SFF backup requires firmware 6.21.0 or greater. HSMs with older firmware do not support SFF backup.

The SFF backup feature can be added only to PED-authenticated cloning HSMs. Cloning and SFF backup are two different HSM features that provide copying or archiving of partition objects in different ways, for different purposes. They can co-exist, but with limitations.

Changes to cloning behavior were necessary in order to implement the SFF backup feature on a cloning HSM. These changes come into effect only when an HSM has the SFF backup capability update file (CUF) installed, and the SFF backup feature is turned on in the HSM policies.

An HSM that is factory-configured for cloning supports secure HSM-to-HSM copying of objects. That cloning ability remains part of the HSM throughout its life. An HSM that was configured for cloning before the addition of SFF backup is still capable of cloning, but now additionally can archive objects to off-board storage by means of SFF backup.

A cloning-only HSM (without the SFF capability enabled) can accept cloning only of objects that have never been stored off the HSM (except keys clearly marked as extractable). Therefore, when SFF backup is installed and enabled on a cloning HSM (cloning plus SFF), the operation of cloning to or from that HSM becomes restricted to HSMs that also have SFF backup installed and enabled. This is particularly important in HA implementations. If SFF backup is enabled on an HA group member, it must also be enabled for all other members of the HA group. See ["Effect on HA" on page 89](#) for more information.

Cloning and SFF backup compatibility

The following table sets out the compatibility constraints for HSMs with and without the SFF backup capability.

Source HSM			Target HSM				
Firmware Version	Has CUF?	Has HSM-level policy set? [See Note 1]	Firmware Version	Has CUF?	Has HSM-level policy set? [See Note 1]	Cloning Outcome	SFF backup?
F/w prior to version 6.21.0	N/A	N/A	F/w prior to version 6.21.0	N/A	N/A	No change. Cloning from one HSM to another is possible if the two HSMs share the same cloning domain. This was always the case.	None
F/w prior to version 6.21.0	N/A	N/A	F/w version 6.21.0 or newer	No	No	No change. Cloning from one HSM to another is possible if the two HSMs share the same cloning domain.	None
F/w version 6.21.0 or newer	No	No	F/w prior to version 6.21.0	N/A	N/A	No change. Cloning from one HSM to another is possible if the two HSMs share the same cloning domain.	None
F/w version 6.21.0 or newer	Yes	Yes	F/w prior to version 6.21.0	N/A	N/A	Cloning is NOT possible. Cloning from one HSM to the other is prevented when mismatch of settings is detected.	Source can use SFF backup, Target cannot

Source HSM			Target HSM				
Firmware Version	Has CUF?	Has HSM-level policy set? [See Note 1]	Firmware Version	Has CUF?	Has HSM-level policy set? [See Note 1]	Cloning Outcome	SFF backup?
F/w version 6.21.0 or newer	Yes	Yes	F/w version 6.21.0 or newer	No	No	Cloning is NOT possible. Cloning from one HSM to the other is prevented when mismatch of settings is detected.	Source can use SFF backup, Target cannot
F/w version 6.21.0 or newer	Yes	Yes	F/w version 6.21.0 or newer	Yes	Yes	Cloning from one HSM to another is possible if the two HSMs share the same cloning domain.	Source and Target can both use SFF backup. Can interchange provided the same SIM secret is on both HSMs

Note 1: The partition SFF backup policy does not have an effect at this level. The HSM-level policy governs. The partition policy is used when the HSM-level policy is on and the SO wishes to disallow SFF backup for just a particular partition.

Note 2: In addition to the requirement for minimum firmware level, the Capability Update **must** be present and the appropriate policy **must** be set for the feature to work. The above table has separate columns for each condition to highlight them, but does not include possible instances where the CUF is installed but the policy is off. If any of the three (firmware, CUF, policy) is not correct, the SFF backup feature cannot work.

SFF Backup Compatibility Summary

The following rules apply to the SFF backup feature:

- If your HSM is not factory configured for cloning, you cannot apply the SFF backup capability.
- If your HSM has firmware lower than 6.21.0, you cannot apply the SFF backup capability.
- If your HSM has version 6.21.0 (or higher) firmware, and is a cloning version HSM, you can apply the SFF backup capability.
 - If you do not apply the capability then the HSM can clone as it always did.
 - If you do apply the capability, but do not switch on the policy, cloning is still not affected.
 - If you do apply the capability, and switch on the policy, you can archive partition objects to an SFF backup eToken. Your ability to clone, however, is restricted to other HSMs that also have the SFF capability applied and the policy switched on.

Cloning Compatibility Summary

This section might seem repetitive, given the previous section, but readers might come to this page from a perspective of wishing to clone, or of wishing to use the SFF feature. Viewed from the cloning perspective, the simple statement regarding SFF is that, in the case where either HSM has the SFF policy enabled, the other must also have it enabled for cloning to function. Otherwise, if you wish to clone between the HSMs, then you must disable the policy on the HSM with the SFF CUF. Without that preparation, a cloning attempt results in an error CKR_DATA_INVALID.

Firmware 6.21.0 was the dividing line, the first firmware that supports the SFF feature; if firmware is earlier, then the SFF capability update cannot be installed. In this table, we show several examples of both pre-6.21 and 6.21-or-newer, in both the Source and Destination positions, to indicate that our testing has covered a variety of situations.

Source	Destination	Cloning Result
Pre-6.21.0 (FW6.2.1)	FW6.22.0 with SFF Off	No error
FW6.22.0 with SFF Off	Pre-6.21.0 (FW 6.0.8)	No error
Pre-6.21.0 (FW6.2.1)	FW6.22.0 with SFF On	CKR_DATA_INVALID error
FW6.22.0 with SFF On	Pre-6.21.0 (FW 6.0.8)	CKR_DATA_INVALID error
FW6.22.0 with SFF Off	FW6.22.0 with SFF Off	No error
FW6.22.0 with SFF On	FW6.22.0 with SFF On	No error
FW6.22.0 with SFF Off	FW6.22.0 with SFF On	CKR_DATA_INVALID error
FW6.22.0 with SFF On	FW6.22.0 with SFF Off	CKR_DATA_INVALID error

The takeaway message, where all involved HSMs are cloning type, is:

- If both the intended cloning source and cloning target/destination have older firmware, which does not allow the SFF capability to be installed, then cloning proceeds with no difficulty, as was always the case.
- If either the source or the destination has older firmware, and the other has newer firmware, but with SFF not turned on, then cloning proceeds with no difficulty.
- If either the source or the destination has older firmware, and the other has newer firmware where SFF is installed and ON, then cloning fails.
- If both the source and the destination have newer firmware but SFF is OFF for both, then cloning proceeds with no difficulty.
- If both the source and the destination have newer firmware and SFF is ON for both, then cloning proceeds with no difficulty.
- If both the source and the destination have newer firmware but SFF is ON for one, but OFF for the other, then cloning fails.

Effect on HA

HSMs that do not have SFF backup enabled, and have previously been able to participate in an HA group, continue to function in HA, even when updated to a firmware version that can support SFF backup. This remains true as long as the other members of the HA group have the previous firmware, or have the newer firmware, but with SFF backup not enabled.

HSMs that have the SFF backup capability applied, and the feature policy switched on, can share an HA group only with other HSMs that have the capability applied and the policy switched on.

Applicability

The above general rules apply at the HSM-wide level. It is not possible to have different settings, affecting the above-described compatibilities, at the partition level. The only partition-level option is to forbid SFF backup for a particular partition while the HSM, as a whole, supports and permits it.

Recovering an eToken 7300 for SFF Backup

The eToken 7300 comes pre-configured for one of two certification types, Common Criteria or FIPS. SFF backup currently supports the Common Criteria version eToken, only.

Common Criteria certified eToken 7300 devices work immediately as SFF backup devices, with no configuration or initialization required (the process of performing a backup includes an init step).

In the event that your eToken 7300 is found to be in an unresponsive state, you can attempt to recover the device with the SafeNet Authentication Client Tool.

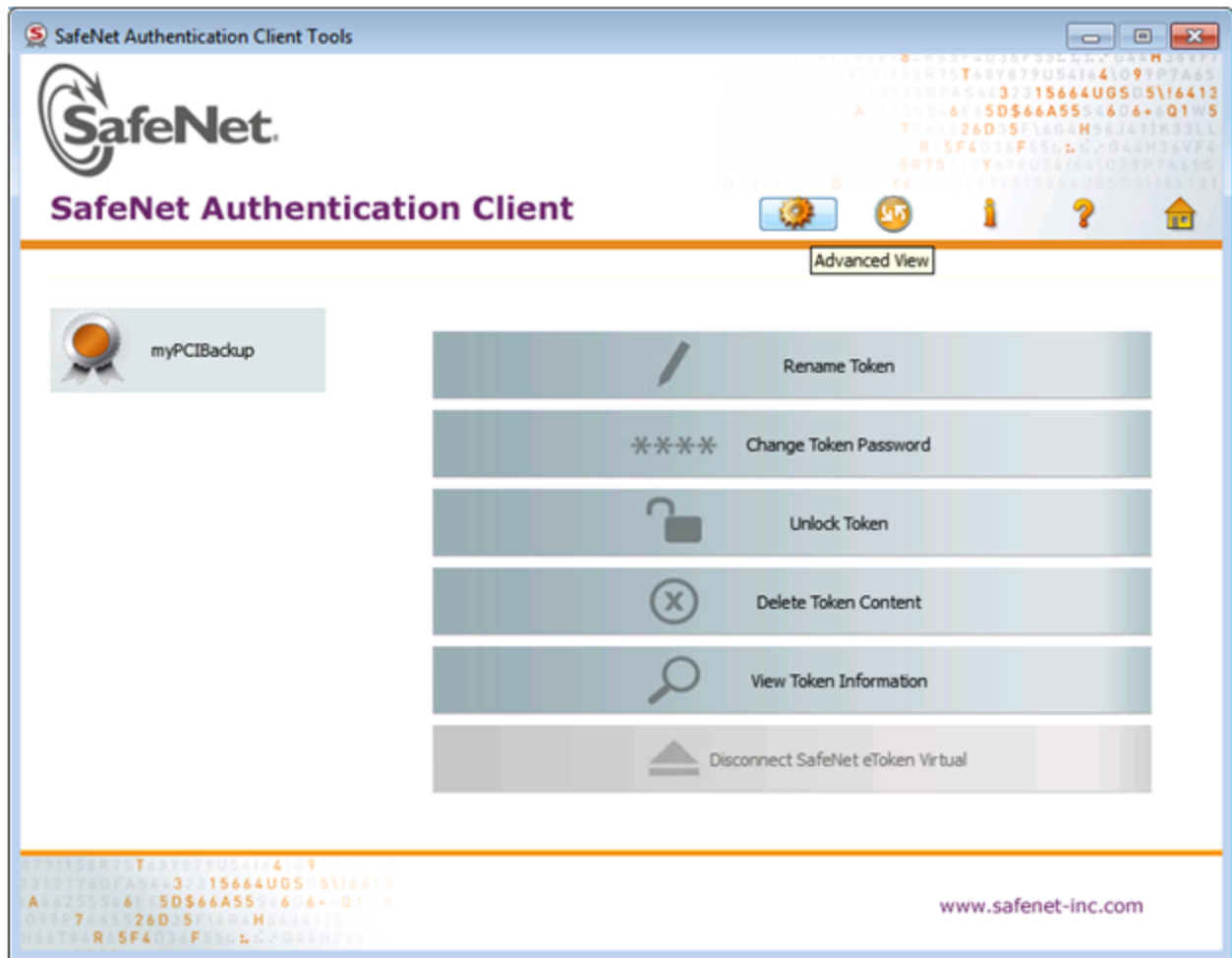
Prerequisites

You will need:

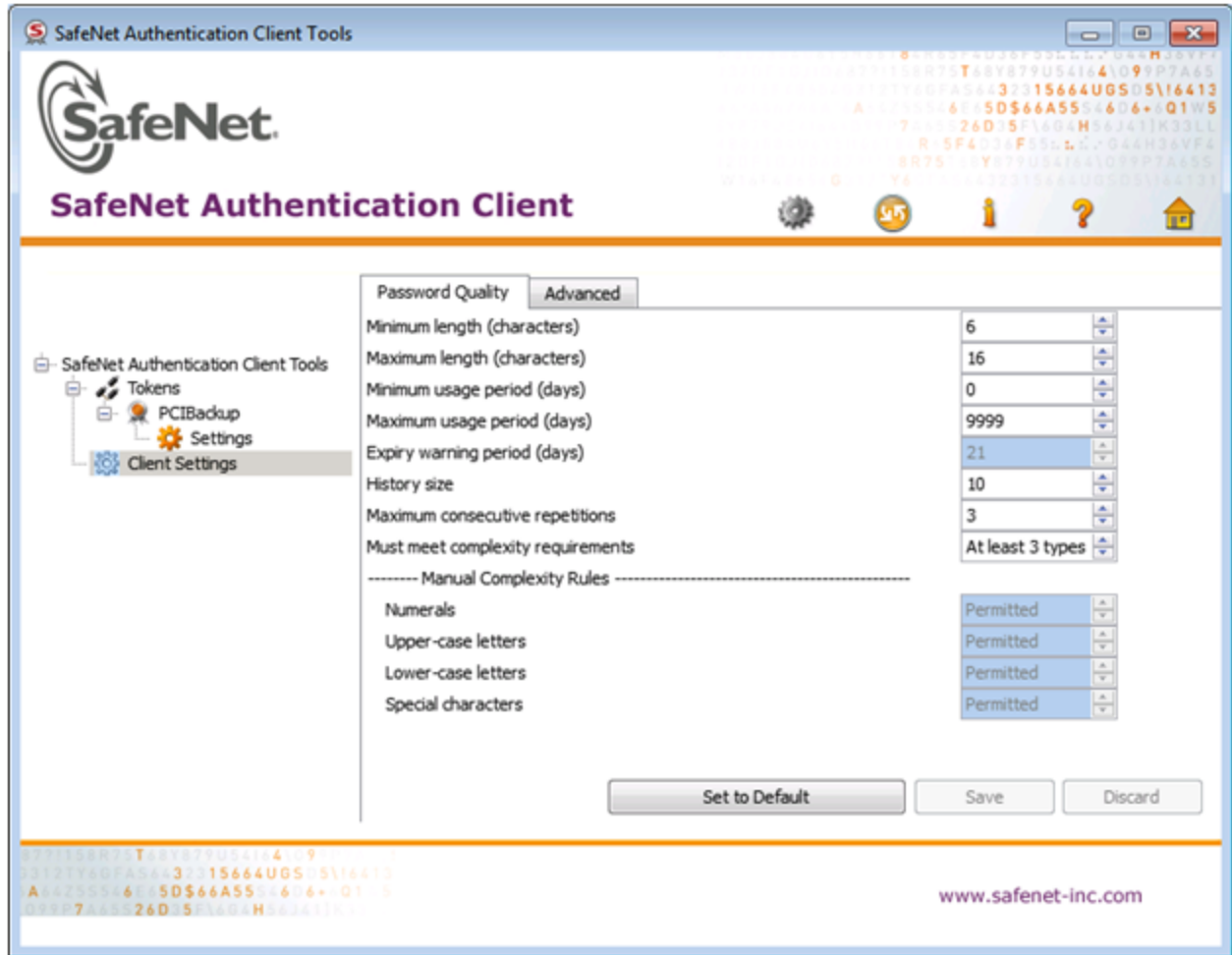
- eToken 7300
- SafeNet Authentication Client Tools software

To recover an eToken

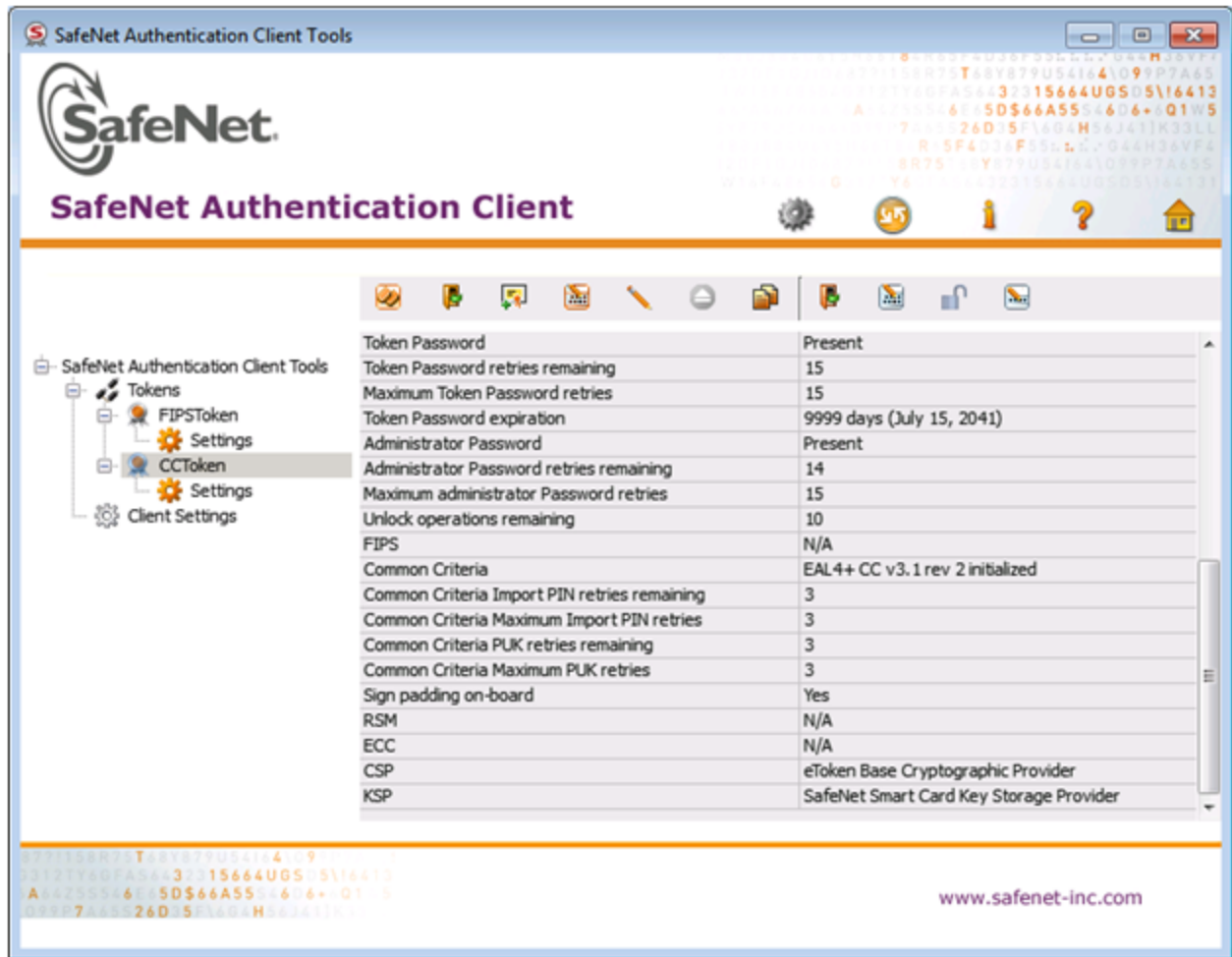
1. Launch **SafeNet Authentication Client Tools** from **Windows > All Programs > SafeNet > SafeNet Authentication Client**, and click the **Advanced View** icon.



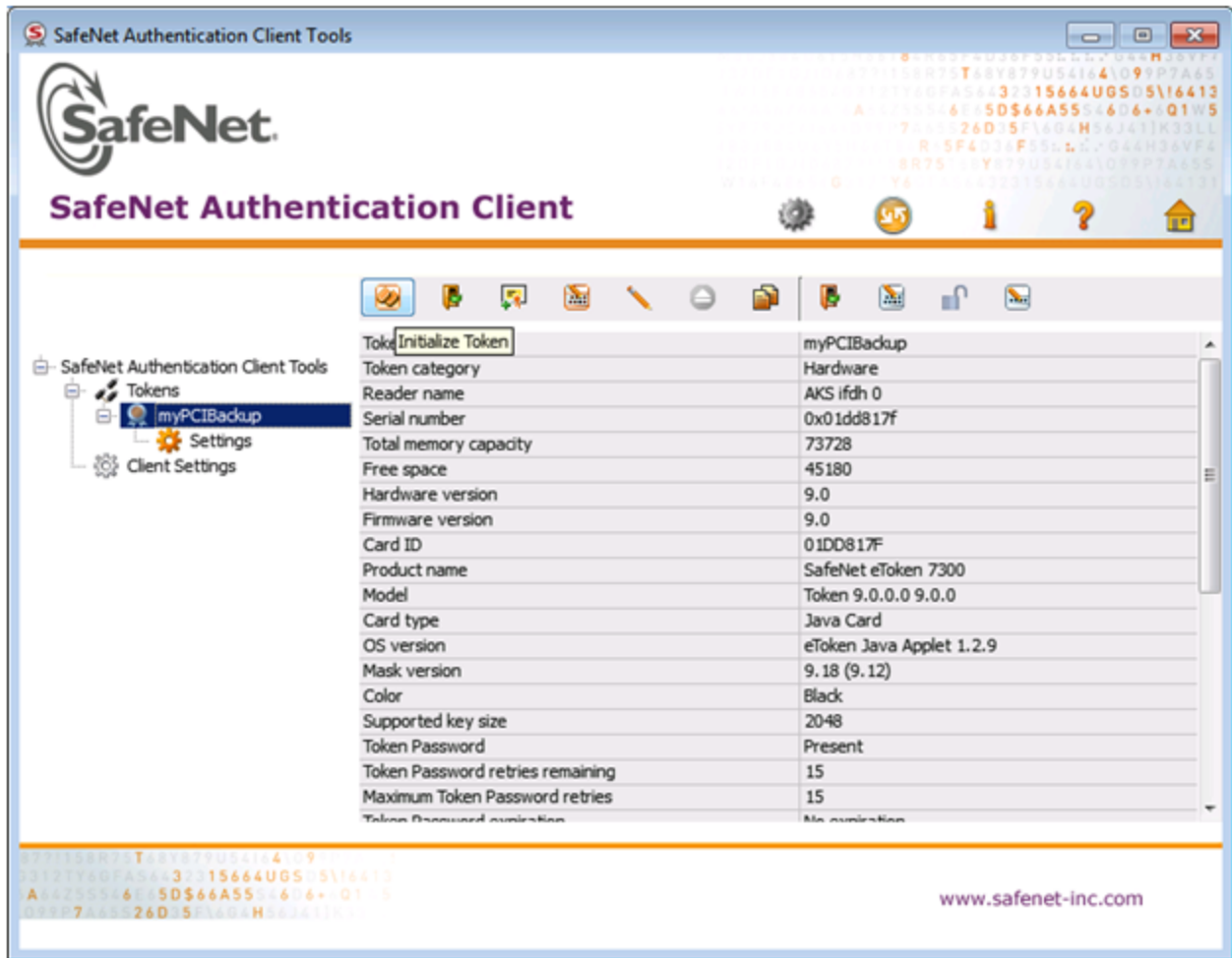
2. Set **Maximum usage period (days)** to 9999 and **Save**.



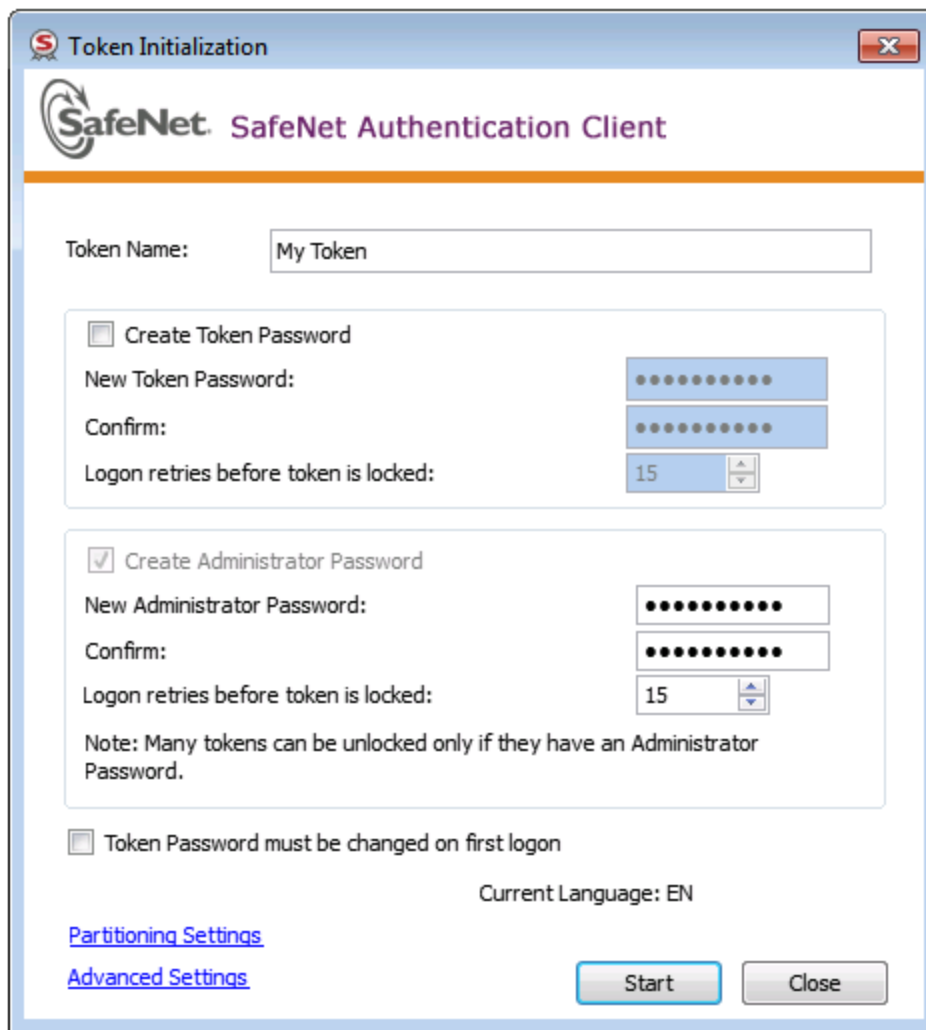
- Confirm that your eToken 7300 is pre-configured as Common Criteria compliant. If that is the case, then in the **CCToken** page, the Common Criteria field says "EAL4+CC v3.1 rev 2 initialized" and the FIPS field says "N/A".



4. Under the **Tokens** heading in the left-hand column, select the eToken you want to initialize.



5. Click the **Initialize Token** icon to start the initialization.
6. On the **Token Initialization** dialog, apply a token name to distinguish this eToken 7300 from other SafeNet SFF backup tokens, and enter a Token Password and an Administrator Password.



The image shows the 'Token Initialization' dialog box of the SafeNet Authentication Client. The window has a title bar with the SafeNet logo and the text 'Token Initialization'. Below the title bar is a header with the SafeNet logo and 'SafeNet Authentication Client'. The main area contains several fields and checkboxes. At the top, there is a 'Token Name' field with the value 'My Token'. Below this, there are two sections. The first section is for creating a token password, with a checkbox 'Create Token Password' (unchecked), a 'New Token Password' field (masked with dots), a 'Confirm' field (masked with dots), and a 'Logon retries before token is locked' spinner set to 15. The second section is for creating an administrator password, with a checkbox 'Create Administrator Password' (checked), a 'New Administrator Password' field (masked with dots), a 'Confirm' field (masked with dots), and a 'Logon retries before token is locked' spinner set to 15. Below these sections is a checkbox 'Token Password must be changed on first logon' (unchecked). At the bottom left, there are two links: 'Partitioning Settings' and 'Advanced Settings'. At the bottom right, there are 'Start' and 'Close' buttons. The current language is set to 'EN'.

Token Initialization

SafeNet SafeNet Authentication Client

Token Name: My Token

☐ Create Token Password

New Token Password:

Confirm:

Logon retries before token is locked: 15

☒ Create Administrator Password

New Administrator Password:

Confirm:

Logon retries before token is locked: 15

Note: Many tokens can be unlocked only if they have an Administrator Password.

☐ Token Password must be changed on first logon

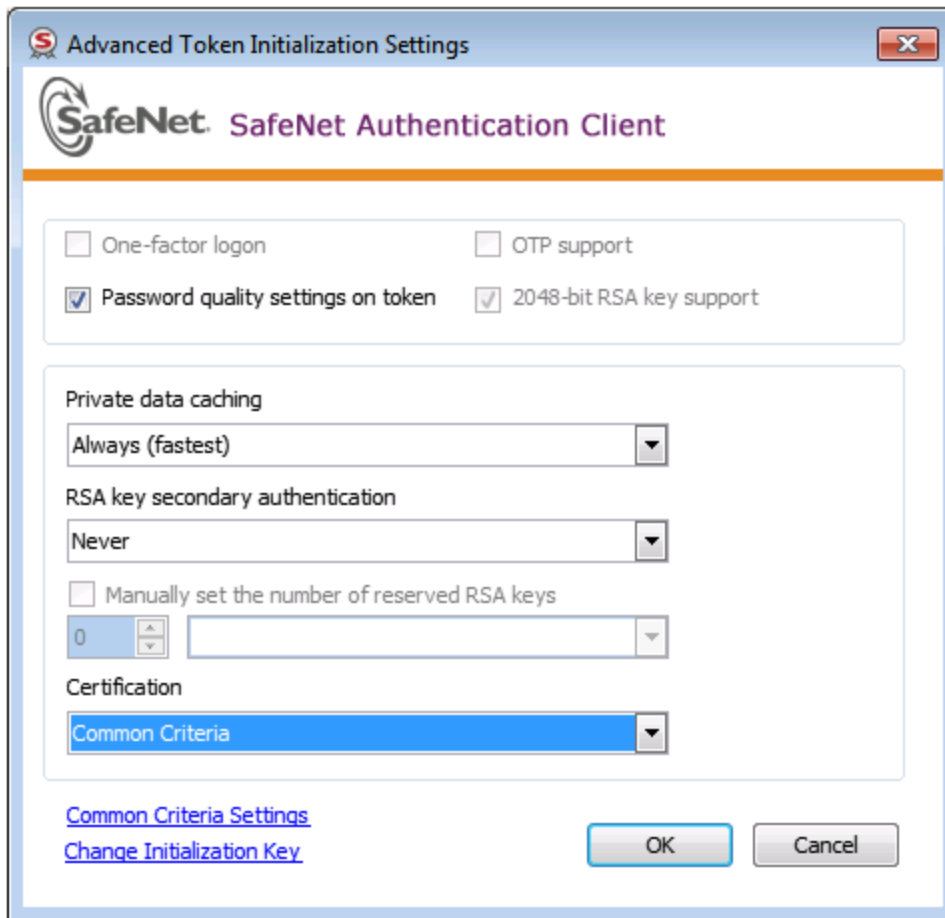
Current Language: EN

[Partitioning Settings](#)

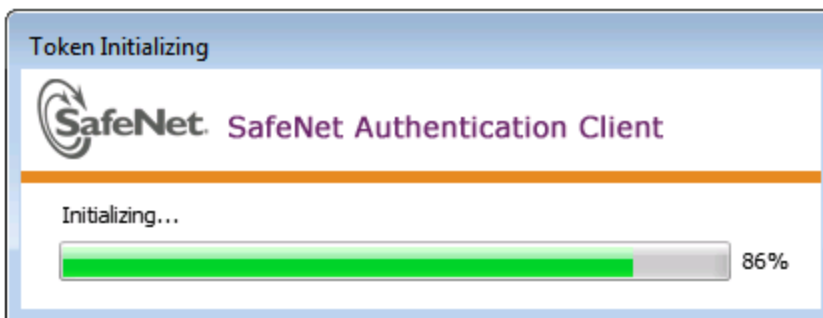
[Advanced Settings](#)

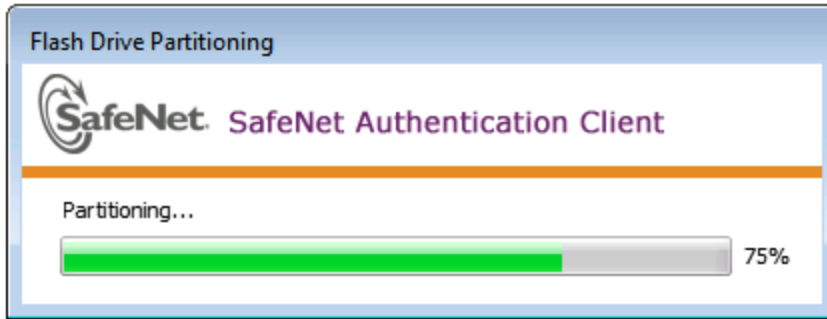
Start Close

7. Select **Advanced Settings** at the bottom left of the dialog.
8. In the **Advanced Settings** dialog, ensure that the **Certification** type matches the type of the eToken (in this case, Common Criteria) and click **OK** to return to the **Token Initialization** dialog.

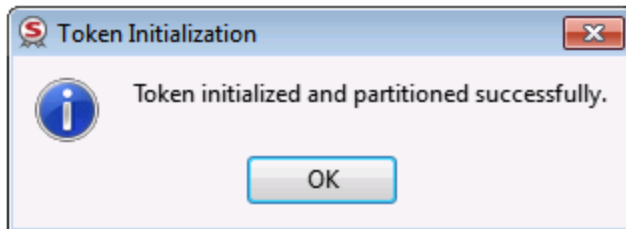


9. In **Token Initialization**, click **Start** to launch token initialization. Two progress bars are shown followed by a success announcement.





Note: If you see the following password-related error, ignore it - passwords are not used in the SFF backup process. Just click [OK] to get to the next dialog.



10. Click **OK** to finish.

If the procedure above succeeds, try using the eToken for backup. If the eToken is still not successfully backing up your HSM partition, contact SafeNet support.

Output of hsm showpolicies After SFF Backup Update

```
[mylunasa] lunash:>hsm showpolicies
```

```
HSM Label:   John
Serial #:    741852
Firmware:    6.21.0
```

The following capabilities describe this HSM, and cannot be altered except via firmware or capability updates.

Description	Value
-------------	-------

=====	=====
Enable PIN-based authentication	Allowed
Enable PED-based authentication	Disallowed
Performance level	15
Enable domestic mechanisms & key sizes	Allowed
Enable masking	Disallowed
Enable cloning	Allowed
Enable 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	Disallowed
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
Enable ECIES support	Disallowed
Enable Single Domain	Allowed
Enable Unified PED Key	Allowed
Enable MofN	Allowed
Enable small form factor backup/restore	Allowed

The following policies are set due to current configuration of this HSM and cannot be altered directly by the user.

Description	Value
=====	=====
PIN-based authentication	True

The following policies describe the current configuration of this HSM and may be changed by the HSM Administrator.

Changing policies marked "destructive" will 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
Force Single Domain	Off	35	Yes
Allow Unified PED Key	Off	36	No
Allow MofN	On	37	No
Allow small form factor backup/restore	On	38	Yes

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

Restoring HSM Partitions From Legacy Tokens

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

Keys (objects) from multiple SafeNet CA4 tokens, SafeNet PCM tokens (Key Export Signing, RA), or SafeNet HSM Backup Tokens (such as would be used to backup the contents of SafeNet Network HSM 4.x partitions) with differing cloning domains can be consolidated onto one SafeNet Network HSM 5.x HSM, where objects from every token HSM are restored onto a partition corresponding to each token (segregated by legacy cloning domain).

Alternatively, you could set up an HA group to include the legacy HSM(s) and the target HSM(s), and use the HA synchronization function. This still requires that the target HSM(s) must have their modern cloning domains associated with the legacy domains of the legacy source HSM(s) in the HA group.



Note: Restore from a legacy backup token is effectively a data migration, and is one-way only. Backups to a token-style HSM is not a supported operation for SafeNet Network HSM 5.x

For detailed key migration procedures, go to the Support portal and search for SafeNet HSM Key Migration instructions.

To restore an HSM partition from a legacy token

1. Connect all the required components and open a terminal session to the SafeNet Network HSM appliance.
2. Open a LunaSH session on the SafeNet Network HSM appliance.

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

```
SafeNet Network HSM 5.1.0-25 Command Line Shell - Copyright (c) 2001-2011 SafeNet, Inc. All
rights reserved.
[myluna] lunash:>
```

3. Use the **token backup update firmware** command to upgrade the firmware on the backup token to the latest version. The firmware is included on the appliance.
4. Create a partition to restore to, if it does not already exist.
5. Use the **partition restore** command to restore a partition, adding to, or replacing the current partition contents:

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

CAUTION: Are you sure you wish to erase all objects in the

```
partition named:                p1
Type 'proceed' to continue, or 'quit' to quit now.
> proceed
Warning:  You will need to attach Luna PED to the SafeNet Backup HSM to complete this operation.

           You may use the same Luna PED that you used for SafeNet Network HSM.

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

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

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

Backing Up and Restoring Your HSM SO Space

HSM backup securely clones the SIM masking key from the SafeNet Network HSM SO space to a Backup HSM. Currently, this is important for SIM applications, to preserve the SIM masking key, and also to allow the SIM masking key to be restored onto another HSM in the case of HA with SIM.

Backup/restore of the SO space is a local operation only, using LunaSH. The Backup HSM must be physically connected to the SafeNet Network HSM appliance. That is, there is no provision to backup a SafeNet Network HSM Admin partition remotely, and LunaCM does not support it.

The authentication type must match - if your source Backup HSM is password authenticated, then its contents can be restored onto a password authenticated HSM only; if your source Backup HSM is PED authenticated, then its contents can be restored onto a PED authenticated HSM only.



Note: The Backup HSM and the target HSM must share the same cloning domain.



Note: The **hsm restore** operation has an option to add material from a backup token to an HSM, rather than to replace any material that is already on the HSM, if that is desired. However, the **hsm backup** operation (from HSM onto token) is an overwrite operation, only.

To backup the SafeNet Network HSM SO space

To backup the SO space on a SafeNet Network HSM, have ready a SafeNet Remote Backup HSM, connected to the front-panel USB port of the SafeNet appliance.

1. Login to the SafeNet appliance as admin.
2. At the `lunash` prompt, type:

Password authentication	hsm backup -password <HSM_Admin_password> -domain <domain_string> -tokenpw <password>
PED authentication	hsm backup

If you see an error message about the token not being in "Factory Reset state", see ["Troubleshooting" below](#).

To restore the SafeNet Network HSM SO space

To restore the SO space on a SafeNet Network HSM, have ready a SafeNet Remote Backup HSM, connected to the front-panel USB port of the SafeNet appliance.

1. Login to the SafeNet appliance as admin.
2. At the `lunash` prompt, type:

Password authentication	hsm restore -serial <backup_hsm_serialnum> -password <hsm_admin_password> -tokenadminpw <token_password>
PED authentication	hsm restore -serial <backup_hsm_serialnum>

Troubleshooting

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

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

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

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

```
Warning: This token is not in the factory reset (zeroized) state.
You must present the current Token Admin login credentials
to clear the backup token's contents.
```

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

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

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

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

```
lunash:>token backup factoryreset -serial 1234567
CAUTION: Are you sure you wish to reset this backup token to
factory default settings? All data will be erased.
```

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

```
'quit' to quit now.  
> proceed
```

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

```
lunash:>token backup init -label mylunatoken -serial 1234567 -force  
    Luna PED operation required to initialize backup token - use  
    Security Officer (blue) PED key.  
    Luna PED operation required to login to backup token - use  
    Security Officer (blue) PED key.  
    Luna PED operation required to generate cloning domain on  
    backup token - use Domain (red) PED key.
```

```
'token init' successful.
```

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

Capabilities and Policies

HSMs, and partitions within them, are characterized by capabilities that are set at the factory, or added by means of capability updates, and that are adjusted by means of settable policies that correspond to some of the capabilities.

HSM capabilities, and the HSM policies that derive from them, apply HSM-wide.

Application partition capabilities, and the application partition policies that derive from them, can be inherited from the HSM, or control characteristics that make sense only at the application partition level.

All policies have an equivalent capability, but not all capabilities are matched by a policy that allows adjustment of the capability.

Some policy settings are numerical values that can be increased or decreased. Most policy settings are simply OFF/ON switches. Policy setting requires that the SO be logged in. For HSM-wide policies, and for partition policies in legacy partitions, that is the HSM SO. For partition-level policies in PPSO partitions, that is the partition SO.

Set policies with the `hsm changepolicy` command or the `partition changepolicy` command, as appropriate. The command requires that you identify the policy number that is to change, and the new value it is to hold. For OFF/ON policies, the value is set as zero or one, respectively.

The following sections list and describe the Capabilities and (if it exists) the relevant Policy that corresponds with each.

See ["HSM Capabilities and Policies"](#) below.

See ["Partition Capabilities and Policies"](#) on page 109.

HSM Capabilities and Policies

HSM capabilities represent pre-set or designed-in capacities of the HSM, and are displayed using the **hsm showpolicies** command. Policies correspond to capabilities, and represent modifications that you can apply to any capability that has a corresponding policy (some do not). The command displays the currently-applied capabilities, and then displays the currently available HSM Policies and their values.

Partition capabilities are inherited from the HSM capabilities and policies (where applicable) and, they too can be adjusted by means of partition policies.

The list that you see for your HSM depends on the type of HSM. As well, capabilities might be added if you purchase and apply a capability update to enhance your HSM.

If a capability can be modified by a policy setting, then the change is always in the direction of greater security. A policy can never relax the level of security that is set by a capability.

In some cases, a setting change must force the wiping of a partition or of the entire HSM as a security measure. Those policies are listed as "destructive". The table below summarizes the relationships and provides a brief description of the purpose and operation of each capability and policy.

To reset the policies to their default values

With firmware 6.22.0, or later, you can use the command **"hsm factoryreset"** on page 1 in the *LunaCM Command Reference Guide* to zeroize the HSM and reset the policies to their default values.

With pre-6.22.0 firmware, the **hsm factoryreset** command does not reset the policies, and they remain as configured prior to the command being invoked.

HSM Capability Name	HSM Policy Name	Destructive	Modifiable	Description
Enable PIN-based authentication	Allow PIN-based authentication	-	No	If allowed, use keyboard for entering passwords. (The HSM Admin may never modify the corresponding policy directly. The policy is set during initialization of the HSM.)
Enable PED-based authentication	Allow PED-based authentication	-	No	If allowed, use the SafeNet PED (as well as the keyboard) for entering passwords (via PED Keys). The HSM Admin may never modify the corresponding policy directly. The policy is set during initialization of the HSM.
Performance level	-	-	-	Indicates the performance level of this HSM. The HSM Admin may never modify this capability - it has no corresponding policy. Possible levels are 15: max performance ~7000 1024-bit RSA sigs/sec 4: ~ 1700 1024-bit RSA signatures per second
Enable domestic mechanisms & key sizes	-	-	-	If allowed, this SafeNet HSM is capable of full strength cryptography (i.e. no US export restrictions)
Enable masking	Allow masking	Yes	Yes	If allowed, the SafeNet HSM is capable of SIM, and this feature can be turned on or off by the HSM Admin. If not allowed, the SafeNet HSM is not capable of SIM, and there is no way for the HSM Admin to change this. Needed for Small Form Factor backup.
Enable cloning	Allow cloning	Yes	Yes	If allowed, the SafeNet HSM is capable of backup to Backup tokens, and this feature can be turned on or off by the HSM Admin. If not allowed, the SafeNet HSM is not capable of backup and there is no way for the

HSM Capability Name	HSM Policy Name	Destructive	Modifiable	Description
				HSM Admin to change this. Partition backup or partition network replication is allowed for the SafeNet high availability feature.
Enable special cloning certificate	-	-	-	If allowed, this SafeNet HSM can have a vendor-specific cloning certificate loaded on to it. This policy is always set to not allowed on current SafeNet HSMs.
Enable full (non-backup) functionality	-	-	-	If allowed, this SafeNet HSM can perform cryptographic functions. This policy is always set to allowed on SafeNet HSMs.
Enable ECC mechanisms	-	-	-	If allowed, new changes to existing licenses may be done in the field. This policy is always set to not allowed on SafeNet HSMs.
Enable non-FIPS algorithms	Allow non-FIPS algorithms	yes	yes	If allowed, the SafeNet HSM permits use of cryptographic algorithms that are not sanctioned by the FIPS 140-2 standard, the HSM Admin can select whether to permit use of those algorithms or to adhere to strict FIPS 140-2 regulations. If not allowed, the SafeNet HSM will only operate with FIPS 140-2 approved algorithms, there is no way for the HSM Admin to change this.
Enable SO reset of partition PIN	SO can reset partition PIN	Yes	Yes	If allowed, the SafeNet HSM has the ability to either lock out users or erase them upon X consecutive bad login attempts, if the HSM Admin sets the corresponding HSM policy to "on", users will be locked out and the HSM Admin can reset their password, if the HSM Admin sets the policy to "off", users will be erased after X consecutive bad login attempts. If this capability is not allowed, the SafeNet HSM will always erase users after X consecutive bad login attempts, the

HSM Capability Name	HSM Policy Name	Destructive	Modifiable	Description
				HSM Admin may not change this.
Enable network replication	Allow network replication	No	Yes	If allowed, the SafeNet HSM may use the SafeNet high availability feature, and the HSM Admin may turn this feature on or off. If not allowed, the SafeNet HSM is not capable of automatic network replication for high availability. Partition backup or partition network replication is allowed for the SafeNet high availability feature. (Does not apply to SafeNet PCI.)
Enable Korean Algorithms		No	Yes	If allowed, the SafeNet HSM may use the Korean algorithm set.
FIPS evaluated	HSM has been evaluated and validated to FIPS 140 -2 (or 3)	No	No	Deprecated - no longer used.
Manufacturing Token	-	-	-	N/A (SafeNet internal use, only)
Enable Remote Authentication (*)	Allow Remote Authentication	Yes	Yes	(* Deprecated - Remote Admin and Remote Authentication are no longer supported. The feature is replaced by Remote PED.)
Enable forcing user PIN change	Force user PIN change after set/reset	No	Yes	If allowed, forces the Partition User to perform a <code>partition changePw</code> operation whenever the SO resets the User password (or creates the User Partition). That is, the User cannot perform any other actions on the Partition until the password change is completed. The purpose is to maintain the separation of roles between the SO/HSM Admin and the Partition User/Owner.
Enable portable masking key	Allow off-board storage	No	Yes	Allows or disallows the use of the portable SIM key.
Enable partition groups	Allow partition groups	No	No	Deprecated - not supported.
Enable Remote PED	Allow remote PED	No	Yes	Allow authentication via remotely

HSM Capability Name	HSM Policy Name	Destructive	Modifiable	Description
usage	usage			located SafeNet PED 2 (Remote Capable) and pedServer.
Enable external storage of MTK split	Not directly modifiable by user	-	-	Allows one of the splits of the MTK, the Secure Recovery Vector, to be stored outside the HSM on a purple Secure Recovery PED Key. Used for Secure Transport Mode, and for controlled/supervised recovery from tamper events. The policy associated with this capability is set automatically when the lunash command "hsm srk enable" is run. If that command is never run, or if the HSM is a password-authenticated version, then both MTK splits remain inside the HSM and recovery from tamper is automatic after restart.
HSM non-volatile storage space	Not directly modifiable by user	-	-	Shows the factory-set amount of non-volatile storage that is available on the HSM.
Enable Acceleration	Allow acceleration	Yes	Yes	This capability controls the mechanisms available within the HSM for key generation (RSA, DSA, KCDSA), and HAM. With the "Allow acceleration" policy switched ON, your application can choose from the full range of mechanisms supported by the HSM, for optimum performance with your application.
Enable Unmasking	Allow unmasking	Yes	Yes	If you "ALLOW" masking & unmasking on the HSM module(s) and the partition(s) "Private & Secret" keys you can securely migrate keys within a single appliance. where partition cloning domains match. If you "ALLOW" cloning on multiple appliances that also have masking & unmasking "ALLOWED" on the HSM (s) and partition(s) "Private & Secret" keys, then you can securely migrate keys with multiple appliances on the same domain.

HSM Capability Name	HSM Policy Name	Destructive	Modifiable	Description
Enable FW5 compatibility mode	-	-	-	Not applicable to SafeNet general-purpose HSMs.
Maximum number of partitions				Shows the maximum number of application partitions that can be created on the HSM, according to factory-installed, or purchased and installed, capability upgrade.
Enable ECIES support	Allow ECIES			Elliptic Curve Integrated Encryption Scheme is enabled by a purchased Capability Update. When the CUF is applied, a Policy setting becomes available to switch ECIES off and on. This is a non-FIPS algorithm. If Allow non-FIPS algorithms is set to ON, that setting overrides this one.
Enable Single Domain	-	-	-	Not applicable to SafeNet general-purpose HSMs.
Enable Unified PED Key	-	-	-	Not applicable to SafeNet general-purpose HSMs.
Enable MofN	-	-	-	Not applicable to SafeNet general-purpose HSMs.
Enable small form factor backup/restore				A purchased capability update enables this capability - backup the contents of an HSM partition to a SafeNet eToken 7300, by means of a SafeNet PED. Requires that Masking be enabled and allowed.
Enable Secure Trusted Channel	Allow Secure Trusted Channel			As an HSM policy, this setting enables the use of STC by the application partitions, but does not force it. As an application partition policy, the use of STC can be turned on, or not, for the individual application partition, but only if the HSM-wide policy is set to ON.
Enable decommission on tamper				Not applicable to SafeNet general-purpose HSMs.

HSM Capability Name	HSM Policy Name	Destructive	Modifiable	Description
Enable Per-Partition SO				Enables the capability, HSM-wide, for partitions to be created that have their own Security Officers.
Enable partition re-initialize				Not applicable to SafeNet general-purpose HSMs.

Partition Capabilities and Policies

HSM capabilities represent pre-set or designed-in capacities of the HSM, and are displayed using the **hsm showpolicies** command. Policies correspond to capabilities, and represent modifications that you can apply to any capability that has a corresponding policy (some do not). The command displays the currently-applied capabilities, and then displays the currently available HSM Policies and their values.

Partition capabilities are inherited from the HSM capabilities and policies (where applicable) and, they too can be adjusted by means of partition policies.

The list that you see for your HSM depends on the type of HSM. As well, capabilities might be added if you purchase and apply a capability update to enhance your HSM.

If a capability can be modified by a policy setting, then the change is always in the direction of greater security. A policy can never relax the level of security that is set by a capability.

In some cases, a setting change must force the wiping of a partition or of the entire HSM as a security measure. Those policies are listed as "destructive". The table below summarizes the relationships and provides a brief description of the purpose and operation of each capability and policy.

Partition Capability Name	Partition Policy Name	Modifiable	Description
Enable private key cloning	Allow private key cloning	depends	If this is allowed, the private keys on the partition may be backed up, the HSM Admin can turn this feature on or off. The value of this capability depends on the HSM capability and policy "Enable cloning". If this is not allowed, private keys on this partition cannot be backed up and the HSM Admin may not change this. Partition backup or partition network replication is allowed for the SafeNet high availability feature.
Enable private key wrapping	Allow private key wrapping	depends	If this is allowed, private keys on the partition may be wrapped, and the HSM Admin can turn this feature on or off. If not allowed, private keys on the partition may not be wrapped off. This value is always set to Disallowed for all partitions

Partition Capability Name	Partition Policy Name	Modifiable	Description
			on a SafeNet HSM.
Enable private key unwrapping	Allow private key unwrapping	depends	If this is allowed, private keys may be unwrapped onto the partition, and the HSM Admin can turn this feature on or off. If not allowed, private key unwrapping is not available, and the HSM Admin cannot change this.
Enable private key masking	Allow private key masking	depends	If this is allowed, keys on the partition can use SIM and the HSM Admin can turn this feature on or off. Encryption for this feature uses an AES 256-bit key. The value of this capability depends on the HSM capability and policy "Enable masking". If this is not allowed, this partition cannot participate in SIM, and the HSM Admin cannot change this.
Enable secret key cloning	Allow secret key cloning	depends	If this is allowed, secret keys on the partition can be backed up, and the HSM Admin can turn this feature on or off. (i.e. the HSM Admin may only wish to turn this feature on immediately before a scheduled backup, and then turn it off again to prevent unauthorized backup.) If this is not allowed, secret keys cannot be backed up, and the HSM Admin cannot change this. Partition backup or partition network replication is allowed for the SafeNet high availability feature.
Enable secret key wrapping	Allow secret key wrapping	depends	If this is allowed, secret keys can be wrapped off the partition, and the HSM Admin can turn this feature on or off (i.e. the HSM Admin may wish to not allow secret key wrapping, in which case he/she would set the corresponding policy to "no"). If this is not allowed, the partition does not support secret key wrapping and the HSM Admin cannot change this.
Enable secret key unwrapping	Allow secret key unwrapping	depends	If this is allowed, secret keys can be unwrapped onto the partition, and the HSM Admin can turn this feature on or off. If this is not allowed, the partition does not support secret key unwrapping

Partition Capability Name	Partition Policy Name	Modifiable	Description
			and the HSM Admin cannot change this.
Enable secret key masking	Allow secret key masking	depends	If this is allowed, secret keys on the partition can use SIM, and the HSM Admin can turn this feature on or off. Encryption for this feature uses an AES 256-bit key. If it is not allowed, the partition does not support SIM.
Enable multipurpose keys	Allow multipurpose keys	depends	If this is allowed, keys on the partition may be created for multiple purposes such as signing and decrypting, and the HSM Admin can turn this feature on or off. If not allowed, keys created on (or wrapped onto) the partition must be for single function only. (i.e. specify only one function in the attribute template).
Enable changing key attributes	Allow changing key attributes	depends	If this is allowed, non-sensitive attributes of the keys on the partition are modifiable (i.e. the user can change the functions that the key can use), and the HSM Admin has the ability to turn this feature on or off. If not allowed, keys created on the partition cannot be modified. This policy affects the following "key function attributes": CKA_ENCRYPT CKA_DECRYPT CKA_WRAP CKA_UNWRAP CKA_SIGN CKA_SIGN_RECOVER CKA_VERIFY CKA_VERIFY_RECOVER CKA_DERIVE CKA_EXTRACTABLE All other attributes are not controlled by this policy.
Allow failed challenge responses	Ignore failed challenge responses	depends	If this is allowed, failed challenge responses (HSM Partition Passwords) will not increment the counter for X consecutive bad login attempts, and the HSM Admin can turn this feature on or off. If not allowed, failed challenge responses (HSM Partition Passwords) will increment the failed login counter.

Partition Capability Name	Partition Policy Name	Modifiable	Description
			This capability/policy only pertains to HSMs that use the SafeNet PED for authentication. (The policy name is slightly different from the capability name – if the policy is on, failed challenges are ignored, which is the same as if the capability is allowed.)
Enable operation without RSA blinding	Operate without RSA blinding	depends	If this is allowed, the partition may run in a mode that does not use RSA blinding (Blinding is a technique that introduces random elements into the signature process to prevent timing attacks on the RSA private key. Use of this technique may be required by certain security policies, but it does reduce performance.) and the HSM Admin can turn this feature on or off. If feature is disallowed, the partition will always run in RSA blinding mode; performance will be lower than SafeNet published performance. (The policy name is slightly different from the capability name - if the policy is on, RSA blinding is not used, which is the same as if the capability is allowed.)
Enable signing with non-local keys	Allow signing with non-local keys	depends	If this is allowed, keys that have been wrapped onto the partition may be used (trusted) for signing, and the HSM Admin can turn this feature on or off. If moving keys from software to hardware, this capability must be allowed, and the corresponding policy must be set 'on', or the keys will not be able to perform signing. If not allowed, only keys that were created locally (on the hardware) can be used for signing.
Enable raw RSA operations	Allow raw RSA operations	depends	If this is allowed, the partition may allow raw RSA operations (mechanism CKM_RSA_X_509), the HSM Admin can turn this feature on or off. If not allowed, the partition will not support raw RSA operations.
Max failed user logins allowed	Max failed user logins allowed	depends	The number in the capability indicates the maximum number of consecutive

Partition Capability Name	Partition Policy Name	Modifiable	Description
			failed user logins allowed, as set by the partition license. The HSM Admin can set the corresponding policy to a value less than or equal to the capability value. (i.e. if the capability shows 15, the policy can be set to [1-15], although setting it to a really low number is not recommended.)
Enable high availability recovery	Allow high availability recovery	depends	If this is allowed, another partition that is in high availability mode with this partition may be used to restore login state to this partition after power outage or other deactivation, and the HSM Admin may turn this feature on or off. If not allowed, this partition does not support the SafeNet high availability feature.
Enable activation	Allow activation	depends	If this is allowed, PED Key data for the partition may be cached so subsequent logins do not require PED Keys, and the HSM Admin may turn this feature on or off. If not allowed (or if the policy is turned off) PED Keys must be presented at each login (whether the call is local or from a client application.) This policy only applies to partitions on HSMs that use the SafeNet PED for authentication.
Enable auto-activation	Allow auto-activation	depends	If this is allowed, PED Key data for the partition may be semi-permanently cached to hard disk (encrypted) so that the partition activations status can be maintained after a short power loss, the HSM Admin can turn this feature on or off. If power stays off more than a few minutes, the key that was used to encrypt the data cached to hard disk is no longer valid, so authentication cannot be re-instated. If this capability is not allowed, the partition does not support auto-activation. This policy only applies to partition on HSMs that use the SafeNet PED for authentication

Partition Capability Name	Partition Policy Name	Modifiable	Description
Minimum pin length (inverted: 255 - min)	Minimum pin length (inverted: 255 - min)	yes	<p>The minimum pin length value is determined as follows. Since a policy can only be set to values that are lower (or equal to) the value in a capability, if the min pin length capability was set to 7, the policy could be set to 2, which is a less restrictive policy. This is not acceptable. So, to keep all capabilities consistent, the value of this capability must be interpreted. The formula to use is:</p> $(\text{max pin}) - (\text{min pin}) = (\text{capability value})$ <p>If the minimum pin length capability is set to 248, and the maximum pin length capability is set to 255, the minimum pin is</p> $(255) - (\text{min pin}) = 248 \rightarrow \text{solving for min pin} \rightarrow (\text{min pin}) = 255 - 248 \rightarrow \text{min pin is} \rightarrow 7$ <p>The administrator can set the policy to select a new, more restrictive minimum pin length. Continuing with the example above, assume the administrator wants to set min pin length to 10 to force better password selection. Solve for policy value in the following formula:</p> $(\text{max pin}) - (\text{min pin}) = (\text{policy value}) \rightarrow \text{substituting} \rightarrow 255 - 10 = (\text{policy value}) \rightarrow \text{solving for policy value} \rightarrow \text{policy value is } 245$ <p>To set the minimum pin length to 10, the HSM Admin would change the min pin length policy to 245.</p> <p>Thus, the HSM Admin would select a number less than the capability (245 is less than 255) to set the minimum pin length to a greater value.</p>
Maximum pin length	Maximum pin length	yes	<p>The value here is the maximum value for the pin length. This value is used in calculating the minimum pin length, and the value in the maximum pin length policy always be greater than the value in the minimum pin length policy.</p>
Enable Key Management Functions	Allow Key Management Functions	yes	<p>The HSM Admin or Security Officer can disable access to any key management</p>

Partition Capability Name	Partition Policy Name	Modifiable	Description
			functions by the user - all users become "Crypto-Users" (the restricted-capability user) even if logged in as "Crypto-Officer".
Enable RSA signing without confirmation	Perform RSA signing without confirmation	yes	The HSM can perform an internal verification (confirmation) of a signing operation, in order to validate the signature. By default, that confirmation is disabled because it has a performance impact on signature operations.
Enable Remote Authentication (*)	Allow Remote Authentication	yes	Controls whether the Remote Authentication features can be used at the Partition level ("partition activate" and "partition restore") on a remote SafeNet Network HSM. If this option is switched off but the HSM-level capability is on, then the only Remote Administration tasks that you could perform would be those requiring "hsm login" - no partition-level remote operations. (* Deprecated - Remote Admin and Remote Authentication no longer supported.)
Enable private key unmasking	Allow private key unmasking	Yes	Remove encryption with AES 256-bit key from private key
Enable secret key unmasking	Allow secret key unmasking	Yes	Remove encryption with AES 256-bit key from secret key
Enable RSA PKCS mechanism	Allow RSA PKCS mechanism	Yes	
Enable CBC-PAD (un) wrap keys of any size	Allow CBC-PAD (un) wrap keys of any size	Yes	
Enable private key SFF backup/restore	Allow private key SFF backup/restore	Yes	Small Form-Factor backup/restore is a cloning operation between the current partition and an SFF token. Allow or disallow private keys to be cloned between the partition and the SFF token.
Enable secret key SFF backup/restore	Allow secret key SFF backup/restore	Yes	Small Form-Factor backup/restore is a cloning operation between the current partition and an SFF token. Allow or disallow secret keys to be cloned between the partition and the SFF token.

Partition Capability Name	Partition Policy Name	Modifiable	Description
Enable Secure Trusted Channel	Force Secure Trusted Channel	Yes	Enable the use of Secure Trusted Channel (STC) for the partition. If this is enabled, you have the option to require STC for the current partition, or not.

3

Backup and Restore

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

- "Backup & Restore Overview " below
- "About HSM Backup - Local and Remote" on page 120
- "Backup or SafeNet USB HSM HSM Battery Installation" on page 124
- "Backup your HSM (non-partition) objects" on page 127
- "Backup your HSM Partition" on page 128
- "RBS - Remote Backup Service " on page 129
- "Remote Backup with RBS - the Big Picture" on page 129
- "Prepare RBS to Support Backup / Restore" on page 136
- "Backup your HSM Partition Remotely" on page 138
- "Restore your HSM Partition - locally " on page 149
- "Restore your HSM Partition from Token" on page 150
- "Why is Backup optional?" on page 152
- "How long does data last?" on page 153
- "What is this Error about "token not in factory reset state"?" on page 153
- "Backup HSM Battery Questions" on page 154
- "Small Form Factor Backup" on page 156

Backup & Restore Overview

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

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

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

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

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

- locally, using direct connection at the host
- remotely, via USB connection to a backup workstation with secure network connection to SafeNet HSM's host.

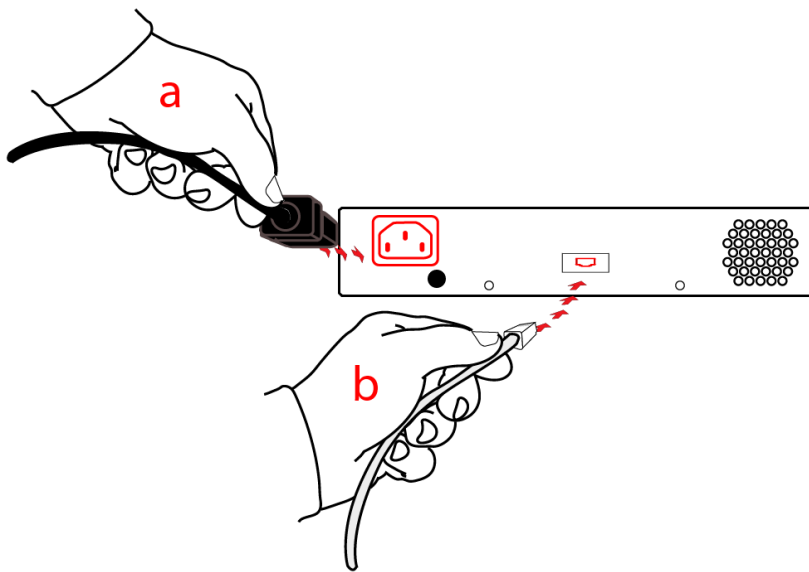
The backup operation looks a lot like the restore operation, because they are basically the same event, merely in different directions.

Connect

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

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

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



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



Source and Target - full or partial

Issue the command "partition backup...".

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

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

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

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

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

Domain

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

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

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

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

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

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

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

Replace or Append

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

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

PED or Password

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

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

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

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

Remote Backup and Restore

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

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

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

About HSM Backup - Local and Remote

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

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

The Backup HSM

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

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

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

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

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

Local Backup of co-located HSMs

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



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

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

The other two backup and restore options:

- Local backup of a distant SafeNet Network HSM
- Remote backup of any SafeNet HSM

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

Local Backup of a Distant SafeNet HSM

This applies only to SafeNet Network HSM, and is not an option for SafeNet PCI-E HSM.

Preparing (configuring) for Remote Backup with Remote PED

While it is standard to remotely administer SafeNet Network HSM, you can also remotely administer an HSM (SafeNet PCI-E HSM embedded in a distant host computer by means of an SSH session or an RDP (Remote Desktop Protocol)

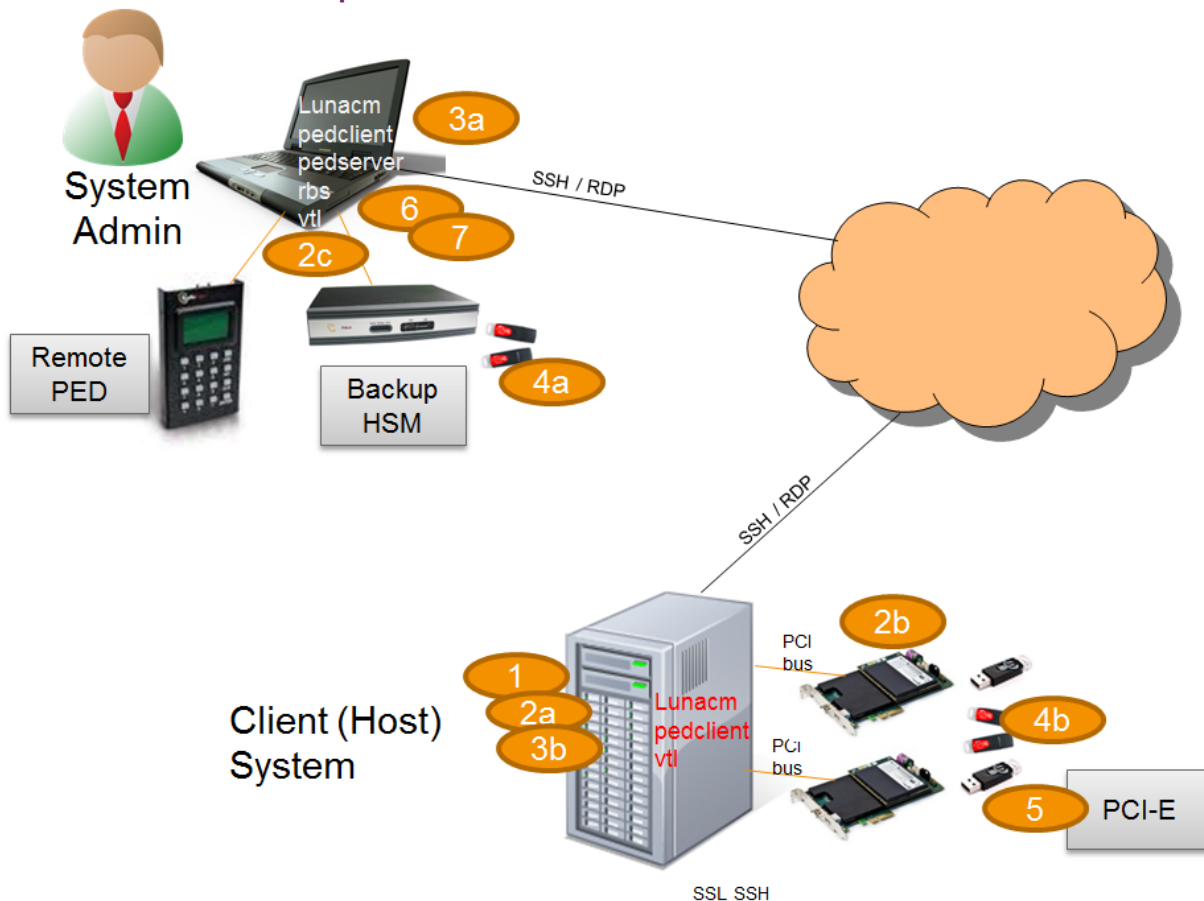
session. You could administer several such HSMs from a central location, including performing backup and restore operations with a SafeNet Remote Backup HSM connected to your Admin computer (perhaps a laptop).

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

Remote Backup

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

Remote Backup of Remote HSM via Remote Client



This scenario is applicable to both SafeNet PCI-E HSM and SafeNet Enterprise HSMs with slight differences in handling.

1	Lunacm is on both the Client (Host) System and the Admin System, but is run on Client (Host) System to launch and manage the backup and restore activity.
2	Lunacm on "Client (Host) System" (2a) sees the primary (2b) and backup (2c) slots and controls the backup/restore.

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

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

Backup or SafeNet USB HSM HSM Battery Installation

The SafeNet Remote Backup HSM (for backing up and restoring HSM and partition contents) and the SafeNet USB HSM (for PKI options) are physically very similar. You could optionally have either, or both, to connect and use with your SafeNet Network HSM.



The battery that powers the NVRAM and RTC in either device is shipped in the packaging, but outside the SafeNet Backup HSM or SafeNet USB 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 SafeNet Network HSM Quick Start Guide).

Installing 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 $\frac{1}{4}$ turn in a counter-clockwise direction. The cover should fall out easily.

3



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

4



Insert the battery, negative end first. The positive end (+) should protrude. The compartment is spring-loaded.

5



Use the battery compartment cover to push the battery into the compartment, against the spring tension. Maintaining the pressure, align the two tabs on the inside of the cover with the two recessed indentations at the top and bottom of the compartment opening. With a little jiggling and a few trial pushes, the tabs should settle into those recesses, allowing the cover to seat flush with the front of the SafeNet USB HSM or SafeNet Remote Backup HSM.

Maintain the inward pressure and twist the cover ¼ turn clockwise to lock it in place. The battery is installed.

6

Replace the front-panel cover by aligning the clips with their respective posts and pushing until the clips grab the posts and the cover snaps in place.

See the main product documentation to connect and use your external HSM with SafeNet Network HSM.

Backup your HSM (non-partition) objects

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

To backup your HSM

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


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

for SafeNet HSM with Password Authentication,

or

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

for SafeNet HSM with PED Authentication.

To restore the HSM contents, reverse the cloning direction.

Disconnecting SafeNet Backup HSM or SafeNet USB HSM

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

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

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

Backup your HSM Partition

Partition backup securely clones partition objects [not including HSM Administrator / SO objects that are contained on the HSM, but not within an HSM Partition] from the HSM Partition, to a SafeNet Backup HSM.

The options are:

- Your SafeNet Backup HSM is connected directly to your HSM's USB port, described below. Use this option when you have just one HSM installed in the host computer.
- Your SafeNet Backup HSM is connected to an administrative computer that is located remotely from the host computer containing your HSM, which is covered separately on ["Backup your HSM Partition Remotely" on page 138](#).

To backup your HSM partition

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

1. Start the lunacm utility.
2. Select the slot to be backed up (if you have more than one HSM installed in the host computer).
3. Login to the source partition as User.
4. At the `lunacm` prompt, type :

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

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

5. The content of the selected partition is copied to the named partition on the directly connected SafeNet Backup HSM.

RBS - Remote Backup Service

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

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

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

Examples of the primary HSM might be:

- a SafeNet PCI-E HSM in its host computer (where the PCI-E HSM is a local slot when viewed by lunacm on the host computer)
- a SafeNet Network HSM partition, seen as a "local" slot in lunacm on a computer that is a registered client of that SafeNet Network HSM.

See "Prepare RBS to Support Backup / Restore" on page 136.

Remote Backup with RBS - the Big Picture

This page depicts some SafeNet HSM concepts around RBS, and Remote Backup and Restore.

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

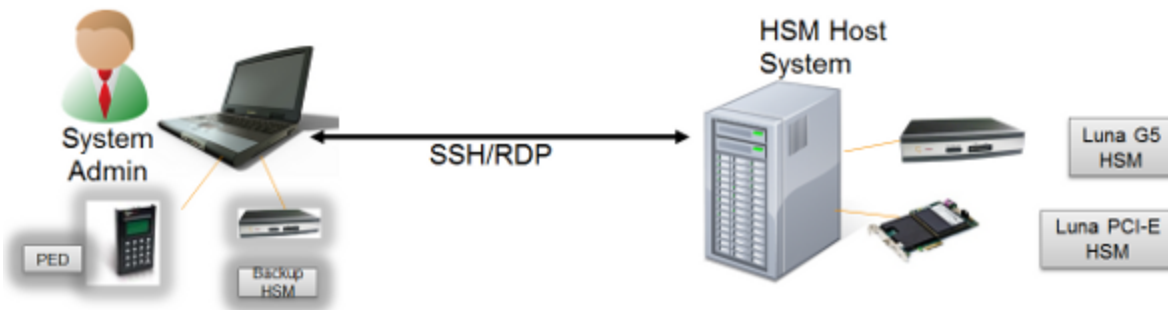
- a host computer with HSM residing in a secure room (server room, or other lock-up with restricted physical access) and
- an administrative workstation, often a laptop with both Remote PED and Remote Backup HSM equipment, communicating with the primary HSM via SSH or Remote Desktop Protocol sessions.

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

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

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

Luna G5 and/or Luna PCI-E in normal operation – not Backup



Admin user and system

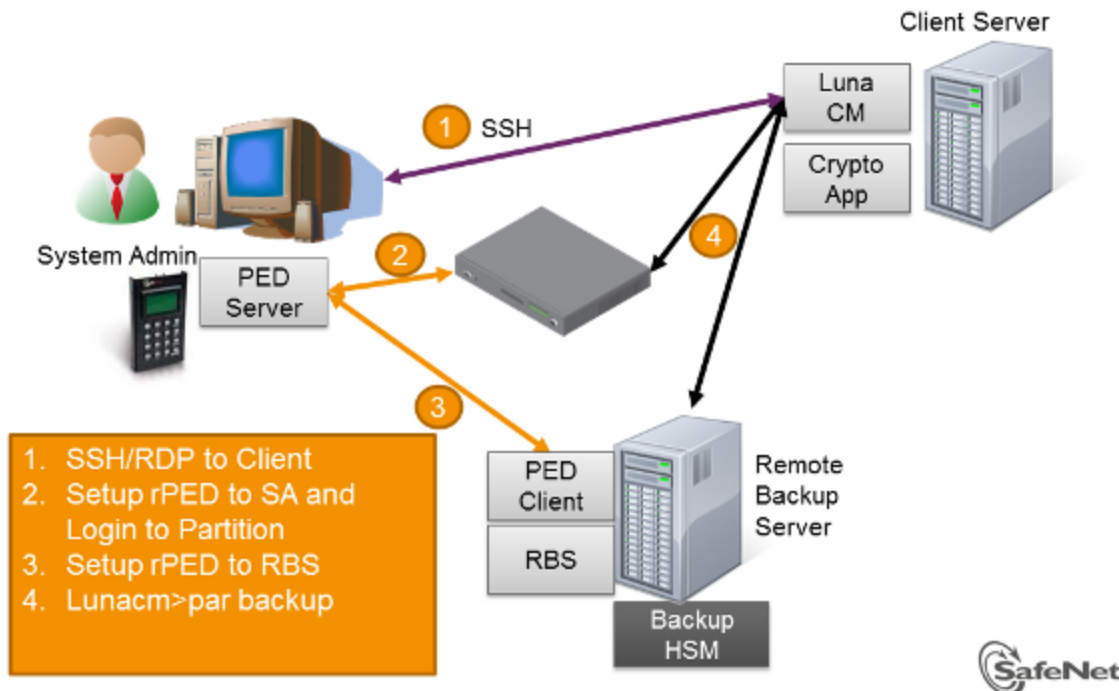
- Takes care of admin duties on one or more distant HSM(s)
- Has **LunaClient** installed, but is not a “client” with access to cryptographic functions on partition

Host Computer System

- Contains G5 HSM, PCI-E HSM
- Has **Lunaclient** installed (mostly for HSM driver)
- Has customer applications installed that use crypto
- Might, or might not service external systems (not shown)

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

Remote Backup – High Level Architecture



Backup is controlled via the `lunacm:>` command line. As a system or security administrator, you choose which computer is to run `lunacm:>` to accomplish the backup/restore operation.

The approach that is chosen comes down to the familiar trade-off between convenience and security.

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

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

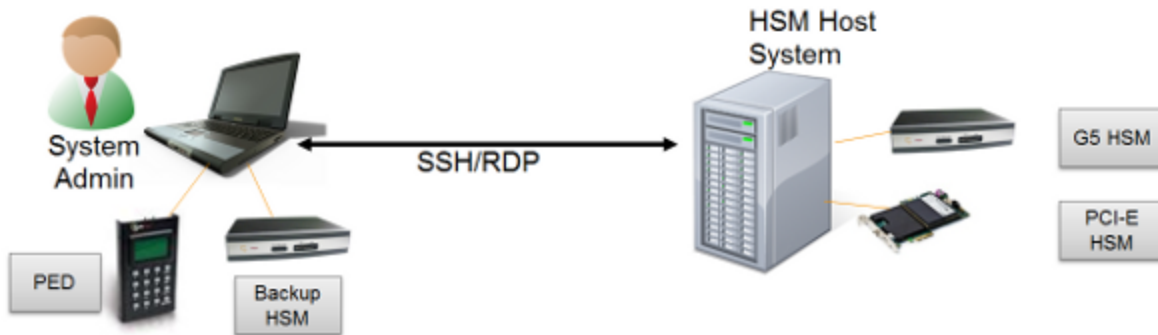
RBS is needed on the Remote Backup computer for this arrangement.

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

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

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

Remote Backup – with Luna G5 and/or Luna PCI-E – part 1



Hardware Configuration

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

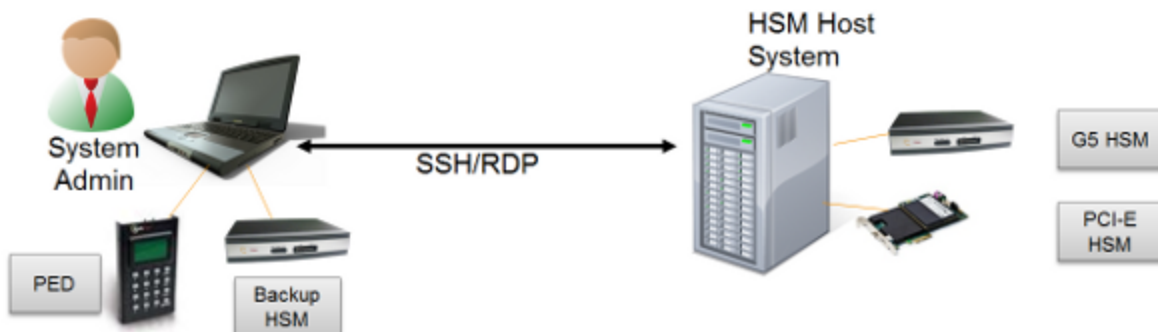
Software Configuration

- Remote Backup Server
- Remote PED Server

Primary HSM Host Computer System

- Luna G5 HSM, Luna PCI-E HSM
- Software Configuration

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



Set up RBS

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

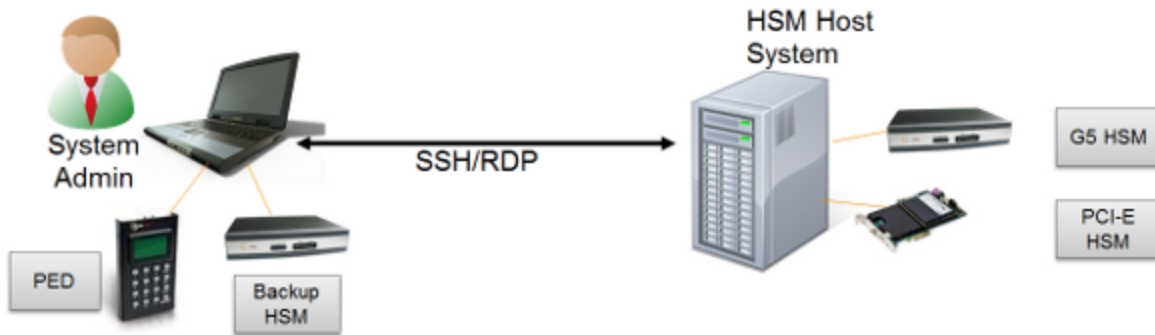
Set up PED Server

- Start PED Server application

Set up HSM Host System

- Add RBS as server using "VTL"
- Start LunaCM

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

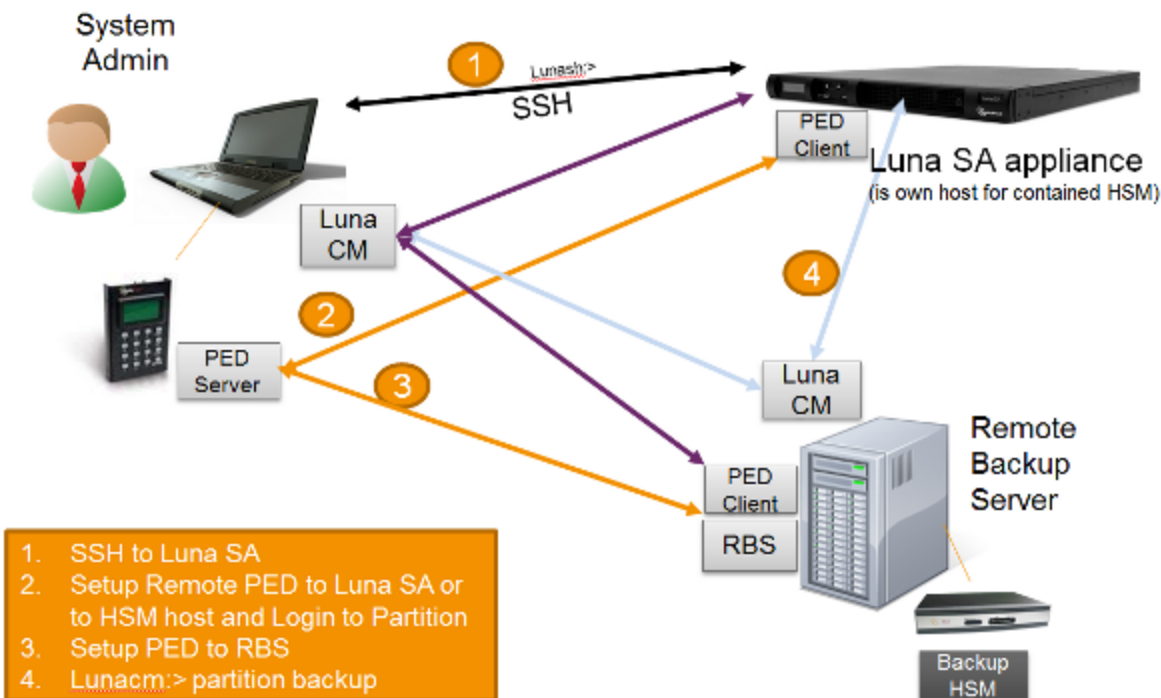


Perform Backup using LunaCM

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

Remote Backup with Remote PED for SafeNet Network HSM, the overview.

Remote Backup – with Luna SA



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

Luna SA in normal operation – not Backup



Admin user and system

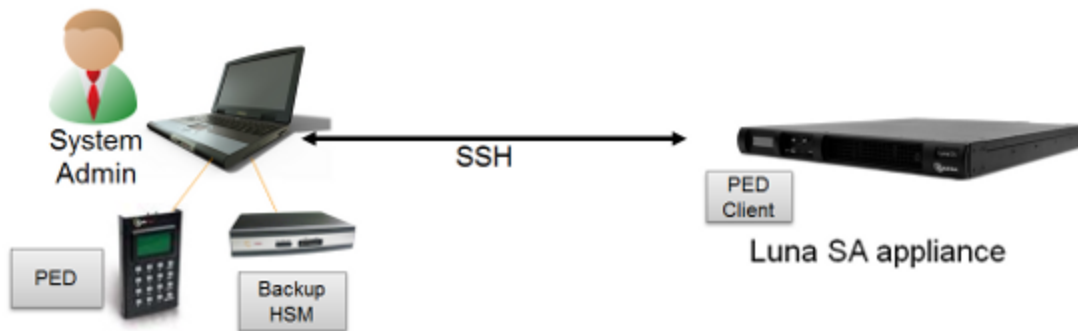
- Takes care of admin duties on one or more distant HSM(s)
- Has **LunaClient** installed, but is not normally a "client" with access to cryptographic functions on partition
- Has **lunacm** and **vti**

Luna SA appliance system

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

Now, a sequence summarizing Remote Backup setup and use.

Remote Backup – with Luna SA – part 1



Hardware Configuration

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

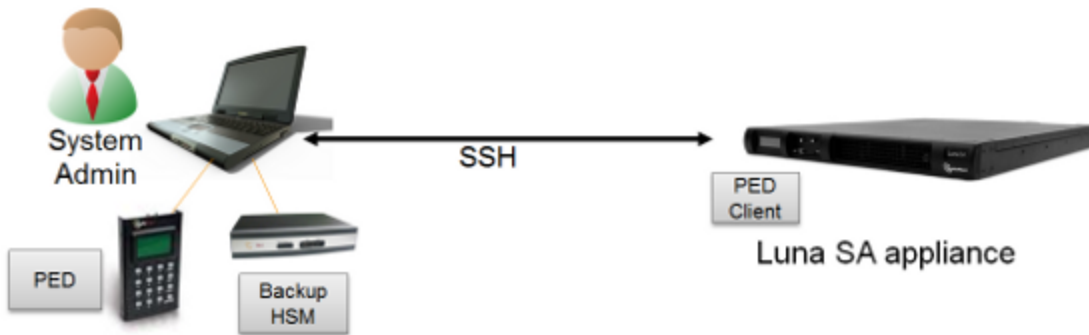
Software Configuration

- Remote Backup Server
- Remote PED Server

Primary HSM (with source partition)

- Luna SA appliance
- Software Configuration

Remote Backup – with Luna SA – part 2



Set up RBS

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

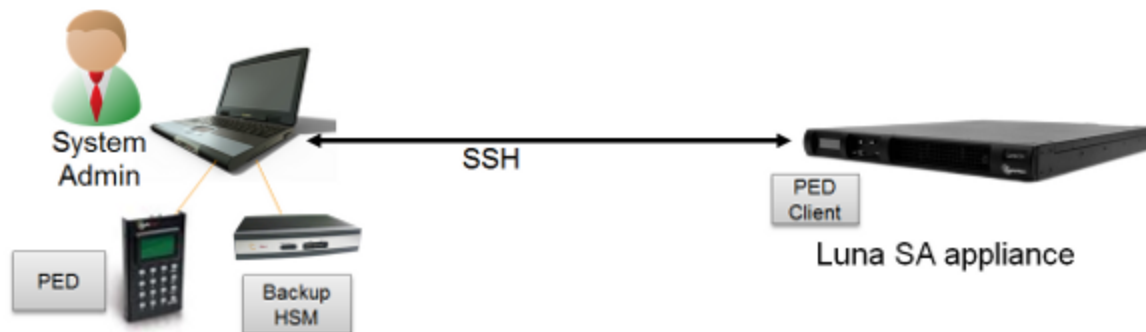
Set up PED Server

- Start PED Server application

Set up HSM appliance

- Add RBS as server using "VTI"
- Start LunaCM on admin/backup station

Remote Backup – with Luna SA – part 3



Perform Backup using LunaCM

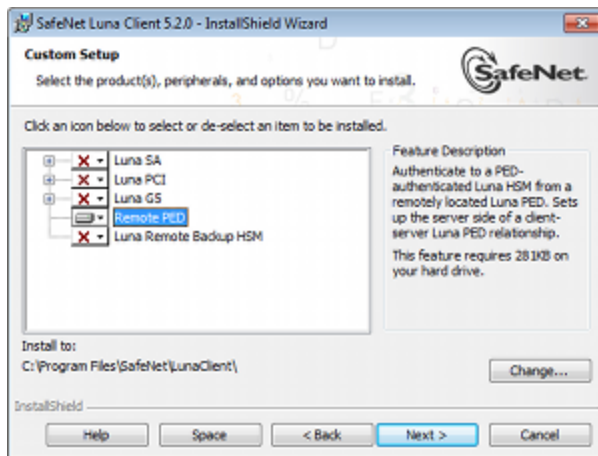
- Set slot to HSM to be backed up
- Connect to the Remote PED
- Login the partition
- Disconnect Remote PED
- Connect Remote PED to remote backup slot
- Execute backup partition command
- Disconnect Remote PED from remote backup slot

Prepare RBS to Support Backup / Restore

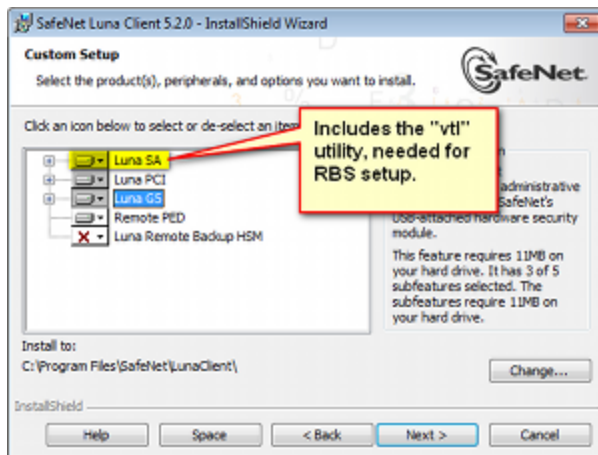
Remote Backup uses the Remote Backup Service (RBS), which must be installed and configured before you use it. RBS is a separate option at software installation time. You do not need it on all client/admin computers, but it doesn't hurt to have it installed.

To prepare for RBS

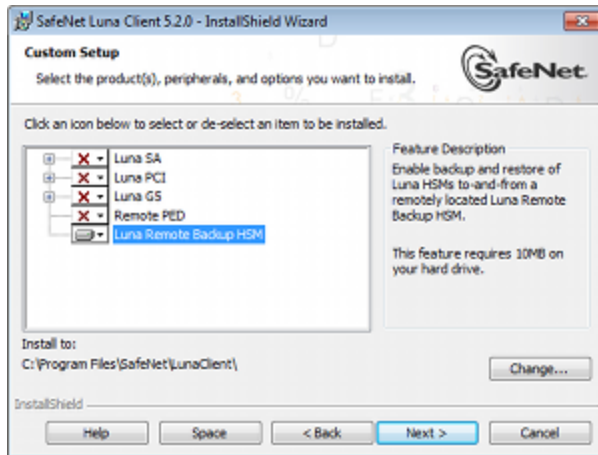
1. Install SafeNet HSM Client software on the computer that will manage your primary HSM (could be the administrative client for SafeNet Network HSM, or the host computer containing one or more SafeNet PCI-E HSMs, or connected to one or more SafeNet USB HSMs). Probably you will want to include the Remote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



If the primary HSM is other than SafeNet Network HSM, install the SafeNet Network HSM option in addition to the SafeNet USB HSM or SafeNet PCI-E HSM software, because the SafeNet Network HSM client is the only one that includes the "vti" utility, necessary for the certificate exchange that enables Remote Backup Service.



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



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

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


```
# scp root@172.20.9.253:/usr/safenet/lunaclient/rbs/server/server.pem .
root@172.20.9.253's password: *****
server.pem | 1 kB | 1.2 kB/s | ETA: 00:00:00 | 100%
```
7. Run "vtl" on the host computer (or appliance) to add the RBS server to the server list.


```
vtl add -n 172.20.9.253 -c server.pem
New server 192.20.9.253 successfully added to server list.
vtl list
Server: 192.20.9.82 HTL required: no
Server: 192.20.9.253 HTL required: no
```

Now go to **"Backup your HSM Partition Remotely"** on the next page.

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

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

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

If you consolidate your HSM administration (including Remote PED) on the same computer with your SafeNet Remote Backup HSM, you would have both PEDClient and PEDServer installed there. We observe that a majority of customers combine administrative functions this way, on a laptop or a workstation that is used to administer one-or-many HSM hosts. The HSM host (with SafeNet USB HSM or SafeNet PCI-E HSM) or the SafeNet Network HSM

appliance resides in a physically secure, possibly remote location, while the administrator works from a laptop in her/his office. Your security policy determines how you do it.

Backup your HSM Partition Remotely

The options to backup a Partition on your SafeNet HSM are:

- local backup
- remote backup.

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

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

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

Remote Backup

You will need -

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

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

Assumptions

The following examples assume that you have set up RBS, as described in ["Prepare RBS to Support Backup / Restore" on page 136](#)

SafeNet Remote Backup HSM and your primary (source) SafeNet HSM are initialized with appropriate keys (blue SO and black Partition Crypto Officer/User PED Keys, which can be the same for both devices, or can be different)

Both devices **must** share the **same** domain or RED key value.

The workstation (Windows computer) has Remote PED and SafeNet Remote Backup software package installed including the appropriate driver.

For SafeNet Network HSM, NTLS is established between your workstation computer, acting as a SafeNet Network HSM client, and the distant SafeNet Network HSM - that is, the workstation is registered as a client with the partition.

Remote PED session key (orange RPV key) has been created and associated with the distant SafeNet HSM.

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

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

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


```
C:\>cd \Program Files\LunaClient
C:\Program Files\LunaClient>PEDServer -mode start
```
5. Open an administrative connection (SSH) to the distant SafeNet HSM (for SafeNet Network HSM appliance, log in as 'admin', for another HSM host, log in with the appropriate ID. Start the PED Client (the Remote PED enabling process on the appliance):

Example (substitute the actual IP address of your workstation computer)--

```
lunash:> hsm ped connect -ip 192.2.12.16 -port 1503
or
lunacm:> hsm ped connect -ip 192.2.12.16 -port 1503
```

Insert the orange RPV PED Key that matches the RPV of the distant SafeNet HSM.

The Remote PED Client in the SafeNet Network HSM appliance or in the SafeNet PCI-E HSM or SafeNet USB

HSM host establishes a connection with the listening PEDserver on your workstation.

6. Proceed to the Backup and Restore examples, below.

RBS Remote Backup with Single Remote PED on Windows

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

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

Backup from a SafeNet Network HSM slot

This example assumes that you have already "Prepare RBS to Support Backup / Restore" on page 136.

That is, briefly:

- You have SafeNet HSM Clientsoftware installed for your primary HSM (source of objects to be backed up).
- You have SafeNet HSM Clientsoftware installed with the RBS option on the host computer connected to your Backup HSM.
- You have run rbs to generate private key/certificate, run rbs again to configure (select device(s) to support), run rbs again to launch the daemon (Linux/UNIX) or the service (Windows).
- You have copied the certificate [server.pem] to your primary HSM host computer (or SafeNet Network HSM appliance).
- You have run vtl on the host computer (or appliance) to add the RBS server to the server list.

To backup from a SafeNet Network HSM slot

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

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

```
LunaCM V2.3.3 - Copyright (c) 2006-2013 SafeNet, Inc.
```

```
Available HSM's:
```

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

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

```
Slot Id -> 3
```



```

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

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

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

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

```

Current Slot Id: 1

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

```
lunacm:> slot set slot 1
```

```
Current Slot Id: 1      (SafeNet Network HSM Slot 6.10.1 (PED) Signing With Cloning
Mode)
```

Command Result : No Error

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

```
lunacm:>ped get
```

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

Command Result : No Error

```
lunacm:> ped connect ip 172.20.10.190
```

Command Result : No Error

```
lunacm:> ped get
```

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

Command Result : No Error

4. [Skip this step if your source partition is Activated]

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

```
lunacm:> par login
```

Option -password was not supplied. It is required.

Enter the password: *****

User is activated, PED is not required.

Command Result : No Error

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

```
lunacm:> ped disconnect
```

Are you sure you wish to disconnect the remote ped?

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

Command Result : No Error

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

Command Result : No Error

```
lunacm:> ped get -slot 3
```

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

Command Result : No Error

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

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

Logging in as the SO on slot 3.

Please attend to the PED.

Creating partition SAbck1 on slot 3.

Please attend to the PED.

Logging into the container SABck1 on slot 3 as the user.

Please attend to the PED.

Creating Domain for the partition SABck1 on slot 3.

Please attend to the PED.

Verifying that all objects can be backed up...

85 objects will be backed up.

Backing up objects...

Cloned object 99 to partition SABck1 (new handle 19).
Cloned object 33 to partition SABck1 (new handle 20).
Cloned object 108 to partition SABck1 (new handle 23).
Cloned object 134 to partition SABck1 (new handle 24).
Cloned object 83 to partition SABck1 (new handle 25).
Cloned object 117 to partition SABck1 (new handle 26).
Cloned object 126 to partition SABck1 (new handle 27).
Cloned object 65 to partition SABck1 (new handle 28).
Cloned object 140 to partition SABck1 (new handle 29).
Cloned object 131 to partition SABck1 (new handle 30).
Cloned object 94 to partition SABck1 (new handle 31).
Cloned object 109 to partition SABck1 (new handle 35).
Cloned object 66 to partition SABck1 (new handle 36).
Cloned object 123 to partition SABck1 (new handle 39).
Cloned object 74 to partition SABck1 (new handle 40).
Cloned object 50 to partition SABck1 (new handle 44).
Cloned object 43 to partition SABck1 (new handle 45).
Cloned object 52 to partition SABck1 (new handle 46).
Cloned object 124 to partition SABck1 (new handle 47).
Cloned object 115 to partition SABck1 (new handle 48).
Cloned object 98 to partition SABck1 (new handle 49).
Cloned object 42 to partition SABck1 (new handle 50).
Cloned object 48 to partition SABck1 (new handle 51).
Cloned object 29 to partition SABck1 (new handle 52).
Cloned object 54 to partition SABck1 (new handle 53).
Cloned object 112 to partition SABck1 (new handle 56).
Cloned object 69 to partition SABck1 (new handle 57).
Cloned object 46 to partition SABck1 (new handle 58).
Cloned object 116 to partition SABck1 (new handle 59).
Cloned object 101 to partition SABck1 (new handle 60).
Cloned object 122 to partition SABck1 (new handle 61).
Cloned object 21 to partition SABck1 (new handle 62).
Cloned object 45 to partition SABck1 (new handle 63).
Cloned object 139 to partition SABck1 (new handle 64).
Cloned object 127 to partition SABck1 (new handle 65).
Cloned object 84 to partition SABck1 (new handle 66).
Cloned object 30 to partition SABck1 (new handle 70).
Cloned object 105 to partition SABck1 (new handle 71).
Cloned object 132 to partition SABck1 (new handle 72).
Cloned object 136 to partition SABck1 (new handle 73).
Cloned object 28 to partition SABck1 (new handle 74).
Cloned object 44 to partition SABck1 (new handle 75).
Cloned object 26 to partition SABck1 (new handle 76).

```

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

```

Backup Complete.

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

Command Result : No Error

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

Restore to a SafeNet Network HSM Slot

If your primary HSM partition (the partition onto which you will restore the backed-up objects) is in Activated state, then only the Backup HSM needs PED activity for authentication during restore. However, we add a couple of steps below

to show that it is straightforward to use the single Remote PED with both HSMs, in the case where your HSM partition is not in Activated state when you begin the restore operation.

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

```
lunacm:> partition clear
```

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

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

```
85 objects were deleted.
```

```
Command Result : No Error
```

```
lunacm:> exit
```

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

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

```
LunaCM V2.3.3 - Copyright (c) 2006-2013 SafeNet, Inc.
```

```
Available HSM's:
```

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

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

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

```
Slot Id -> 4
Tunnel Slot Id -> 6
HSM Label -> PCI422
```

```

HSM Serial Number -> 500422
HSM Model -> K6 Base
HSM Firmware Version -> 6.2.1
HSM Configuration -> Luna PCI (PED) Signing With Cloning Mode
HSM Status -> OK

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

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

```

Current Slot Id: 1

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

```
lunacm:>ped get
```

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

Command Result : No Error

4. Connect to Remote PED.

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

Command Result : No Error

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

```
lunacm:> partition deactivate
```

Command Result : No Error

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

```
lunacm:> partition login
```

```
Option -password was not supplied. It is required.
```

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

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

Command Result : No Error

```
lunacm:> ped disconnect
```

```
Are you sure you wish to disconnect the remote ped?
```

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

```
Command Result : No Error
```

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

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

```
Command Result : No Error
```

```
lunacm:> ped get
```

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

```
Command Result : No Error
```

```
lunacm:> ped get slot 3
```

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

```
Command Result : No Error
```

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

```
lunacm:> partition backup restore -slot 3 -par SABck1
```

```
Logging in to partition SABck1 on slot 3 as the user.
```

```
Please attend to the PED.
```

```
Verifying that all objects can be restored...
```

```
85 objects will be restored.
```

```
Restoring objects...
```

```
Cloned object 19 from partition SABck1 (new handle 20).
Cloned object 20 from partition SABck1 (new handle 21).
Cloned object 23 from partition SABck1 (new handle 22).
Cloned object 25 from partition SABck1 (new handle 25).
Cloned object 24 from partition SABck1 (new handle 26).
Cloned object 26 from partition SABck1 (new handle 28).
Cloned object 28 from partition SABck1 (new handle 29).
Cloned object 27 from partition SABck1 (new handle 30).
Cloned object 29 from partition SABck1 (new handle 33).
Cloned object 30 from partition SABck1 (new handle 34).
Cloned object 31 from partition SABck1 (new handle 40).
Cloned object 35 from partition SABck1 (new handle 44).
Cloned object 36 from partition SABck1 (new handle 45).
Cloned object 39 from partition SABck1 (new handle 48).
Cloned object 40 from partition SABck1 (new handle 49).
Cloned object 44 from partition SABck1 (new handle 53).
Cloned object 45 from partition SABck1 (new handle 54).
```

Cloned object 46 from partition SABck1 (new handle 55).
Cloned object 47 from partition SABck1 (new handle 56).
Cloned object 48 from partition SABck1 (new handle 57).
Cloned object 49 from partition SABck1 (new handle 58).
Cloned object 50 from partition SABck1 (new handle 59).
Cloned object 51 from partition SABck1 (new handle 60).
Cloned object 52 from partition SABck1 (new handle 61).
Cloned object 53 from partition SABck1 (new handle 62).
Cloned object 56 from partition SABck1 (new handle 65).
Cloned object 57 from partition SABck1 (new handle 66).
Cloned object 58 from partition SABck1 (new handle 67).
Cloned object 59 from partition SABck1 (new handle 68).
Cloned object 60 from partition SABck1 (new handle 69).
Cloned object 61 from partition SABck1 (new handle 70).
Cloned object 62 from partition SABck1 (new handle 71).
Cloned object 63 from partition SABck1 (new handle 72).
Cloned object 64 from partition SABck1 (new handle 73).
Cloned object 65 from partition SABck1 (new handle 74).
Cloned object 66 from partition SABck1 (new handle 75).
Cloned object 70 from partition SABck1 (new handle 79).
Cloned object 71 from partition SABck1 (new handle 80).
Cloned object 72 from partition SABck1 (new handle 81).
Cloned object 73 from partition SABck1 (new handle 82).
Cloned object 74 from partition SABck1 (new handle 83).
Cloned object 75 from partition SABck1 (new handle 84).
Cloned object 76 from partition SABck1 (new handle 85).
Cloned object 77 from partition SABck1 (new handle 86).
Cloned object 78 from partition SABck1 (new handle 87).
Cloned object 79 from partition SABck1 (new handle 88).
Cloned object 80 from partition SABck1 (new handle 89).
Cloned object 81 from partition SABck1 (new handle 90).
Cloned object 82 from partition SABck1 (new handle 91).
Cloned object 83 from partition SABck1 (new handle 92).
Cloned object 84 from partition SABck1 (new handle 93).
Cloned object 86 from partition SABck1 (new handle 94).
Cloned object 85 from partition SABck1 (new handle 95).
Cloned object 87 from partition SABck1 (new handle 96).
Cloned object 88 from partition SABck1 (new handle 97).
Cloned object 89 from partition SABck1 (new handle 98).
Cloned object 91 from partition SABck1 (new handle 99).
Cloned object 90 from partition SABck1 (new handle 100).
Cloned object 92 from partition SABck1 (new handle 101).
Cloned object 96 from partition SABck1 (new handle 105).
Cloned object 97 from partition SABck1 (new handle 106).
Cloned object 98 from partition SABck1 (new handle 107).
Cloned object 99 from partition SABck1 (new handle 108).
Cloned object 103 from partition SABck1 (new handle 112).
Cloned object 104 from partition SABck1 (new handle 113).
Cloned object 105 from partition SABck1 (new handle 114).
Cloned object 106 from partition SABck1 (new handle 115).
Cloned object 107 from partition SABck1 (new handle 116).
Cloned object 108 from partition SABck1 (new handle 117).
Cloned object 110 from partition SABck1 (new handle 118).
Cloned object 109 from partition SABck1 (new handle 119).
Cloned object 111 from partition SABck1 (new handle 120).
Cloned object 112 from partition SABck1 (new handle 121).
Cloned object 113 from partition SABck1 (new handle 122).


```

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

```

Restore Complete.

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

Command Result : No Error

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

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

HSM Storage Information for slot 3:

```

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

```

Partition list for slot 3

Number of partition: 1

```

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

```

Command Result : No Error

Restore from backup, using RBS, is complete.

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

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

Restore your HSM Partition - locally

The options, in restoring to a partition are:

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

To restore one HSM Partition, you must have:

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

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

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

1. Insert a SafeNet Backup token into the token-reader slot on the SafeNet appliance front panel.

2. Choose an HSM Partition, and type:

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

Note that in the command above, you could have used `-add` instead of `-replace`.

Example – partition restore Command

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

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

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

Restore your HSM Partition from Token

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

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

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

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

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

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

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

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

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

For SafeNet HSM with Password authentication

1. Create a partition on the SafeNet Network HSM 5 HSM

```
lunash:>partition create -partition mylunapar2 -password <password> -domain <domain> -force
```
2. With the SafeNet DOCK 2 reader powered on and connected (USB) to the SafeNet Network HSM 5.x, insert a SafeNet Network HSM Backup token (or other legacy removable token-format HSM) into the token-reader slot of the SafeNet DOCK.
3. Type the command:

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

For SafeNet Network HSM with PED authentication

1. Create a partition on the SafeNet Network HSM 5 HSM

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

Both user (black) and domain (red) PED keys are created for SafeNet Network HSM 5 partition mylunapar2.
2. With the SafeNet DOCK 2 reader powered on and connected (USB) to the your *client* computer, insert the desired SafeNet Network HSM Backup token or SafeNet CA4 token into the token-reader slot of the SafeNet DOCK 2.
3. Leave the SafeNet DOCK 2 powered on and the token in its slot, and transfer its USB cable connection from the client computer to the USB socket on the SafeNet Network HSM 5.x. The SafeNet Network HSM immediately sees the new token slot, and you can now run lunash commands from the SafeNet Network HSM against the token.
4. Import the legacy domain [SafeNet Network HSM 5, SafeNet USB HSM, and the SafeNet Remote Backup Device use a newer domain scheme, which is not compatible with legacy HSM domains; the partition setLegacyDomain command prepares a legacy domain in a way that allows it to be recognized and used by a current-model HSM, in special circumstances- the HSM retains its modern domain, but the legacy domain becomes associated with the partition's "real" domain. The association is permanent for the life of that partition.

Intentional, designed-in, data security provisions prevent setting/associating a legacy domain from one SafeNet token to a single SafeNet Network HSM 5.x partition, then associating another (different) legacy domain to that

same partition and adding the second token's objects to the partition while the first token's objects are stored there. Just as you cannot clone/copy objects from one token to another token with a different domain, you cannot get around that security provision by migrating unmatched domain objects to a single SafeNet Network HSM partition.

As long as token HSMs share a common (legacy) domain, you migrate the contents of multiple tokens to a single partition - the legacy domain is set just once for all such tokens.] :

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

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

5. Type the command:

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

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

6. Repeat ALL the above steps to restore objects from other SafeNet tokens onto *separate* SafeNet Network HSM partitions.

Repeat only step 6 with the `"-add"` option, instead, to restore objects from other SafeNet tokens onto the *same*, single SafeNet Network HSM 5 partition - this works ONLY if the originating SafeNet tokens all share the same legacy domain. Once a legacy domain is associated with a SafeNet Network HSM 5.x partition, that association remains in force for the life of that partition; the HSM does not allow another association (of legacy domain) to be made onto a partition that already has an existing association. The only way to end the association is to destroy the partition (wiping all contents) and create it again.

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

To backup the HSM contents to a token-style HSM --- this is not a supported operation for SafeNet Network HSM 5.x.

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

Why is Backup optional?

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

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

1. Some Customers don't care. They may be using (for example) SSL within a controlled boundary like a corporation, where it is not a problem to simply tell all employees to be prepared to trust a new certificate, in the event that the previous one is lost or compromised. In fact it might be company policy to periodically jettison old certificates and distribute fresh ones.

Other customers might be using software that manages lost profiles, making it straightforward to resume work with a new key or certificate. The certificate authority that issued the certificates would need backup, but the individual customers of that certificate authority would not.

In summary, it might not be worthwhile to backup keys that are low-cost (from an implementation point of view) to

replace. Keys that carry a high cost to replace should be backed up.

2. Some countries do not permit copying of private keys. If you are subject to such laws, and wish to store encrypted material for later retrieval (perhaps archives of highly sensitive files), then you would use symmetric keys, rather than a private/public key-pair, for safe and legal backup.

How long does data last?

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

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

What does this mean to me?

Advances in technology will probably ensure that you never need to test the expected expiration of data on your SafeNet HSMs.

What is this Error about "token not in factory reset state"?

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

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

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

```
You must present the current Token Admin login credentials
to clear the backup token's contents.
```

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

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

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

```
[mylunasa] lunash:>
```

This is a **security feature**, intended to prevent backup of PED-secured HSM objects onto a less secure Password Authenticated token.

To work around this problem, issue "token factoryReset" first , and then initialize the token:

```
[mylunasa] lunash:>token backup factoryReset -serial 1234567
```

```
CAUTION: Are you sure you wish to reset this backup token to
factory default settings? All data will be erased.
```

```
Type 'proceed' to return the token to factory default, or
'quit' to quit now.
```

```
> proceed
```

```
token factoryReset' successful.
```

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

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

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

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

Luna PED operation required to generate cloning domain on backup token - use Domain (red) PED key.

'token init' successful.

Command Result : 0 (Success)

[mylunasa] lunash:>

Comparison Summary

See "[Comparison of Destruction/Denial Actions](#)" on page 400 to view a table that compares and contrasts various "deny access" events or actions that are sometimes confused.

Backup HSM Battery Questions

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



The battery that powers the NVRAM and RTC must be installed for use, but some questions commonly arise if the device is to be stored for long periods. As an administrator of HSMs, I need clear instructions on what to do/how to manage the battery in the SafeNet USB HSM and SafeNet 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 °C to +85 °C

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 °C

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.

Small Form Factor Backup

The small form factor (SFF) backup feature is available for PED-authenticated SafeNet HSMs only.



Note: A SafeNet PED is required for SFF backup. A SafeNet PED with Remote capability is recommended for SFF backups. See the Customer Release Notes for more information.

Characteristics

Small form factor backup is mediated by SafeNet PED and uses SafeNet eToken 7300 USB devices as the repository for archived cryptographic objects.



- The eToken 7300 is Common Criteria validated and tamper-evident.
- SFF backup is supported for SafeNet Network HSM and SafeNet PCI-E HSM. Small Form-Factor Backup is not supported for SafeNet USB HSM (no capability update file is available). To back up a SafeNet USB HSM, clone the contents to another SafeNet USB HSM.)
- One eToken 7300 can back up one HSM partition.
- Backup and restore can be performed to or from an eToken 7300 inserted into a locally-connected or remotely-connected SafeNet PED (via PedServer).
- A capability update file (CUF) must be purchased and applied to each HSM (serial number specific) that is to use the SFF Backup feature.



Note: Using SFF backup imposes some constraints, and affects other features like HA. See "Cloning and SFF Backup Option Use Cases" on page 86 for more detailed information.



Required Elements

SFF backup requires:

- SafeNet Software version 5.4.0 or newer
- HSM firmware version 6.21.0 or newer
- A SafeNet PED with firmware version 2.6.0-6 or newer. A remote PED is recommended.
- Source SafeNet HSM must be cloning type, only - not applicable to Key Export (KE) HSMs
- Backup to a remotely located SFF backup requires a remote PED - a local-only PED is not field-upgradable to remote capability
- HSM must have the SFF backup capability update applied (this is a purchased option)



CAUTION: The SFF backup capability update is a destructive change to your HSM, meaning that the upgrade enforces HSM initialization and all contents will be lost. Back up any important keys or objects before the upgrade. You can recreate your partition(s) and restore your objects after the HSM has been re-initialized, following the application of the SFF backup capability upgrade.

Configuration

Small Form-Factor Backup requires that the SafeNet configuration file **crystoki.ini** (Windows) or **Chrystoki.conf** (Linux/UNIX) must have two specific settings:

- `CommandTimeOutPedSet = 720000`
- `PEDTimeout2 = 200000`

Newly installed/created SafeNet HSM client configuration files have the necessary entries, with the correct values, but pre-existing clients might be missing an entry or might have an insufficient value assigned.

To configure Linux/UNIX clients

On Linux clients the **Chrystoki.conf** file is saved upon SafeNet HSM client un-installation, and re-used on later installation. Manually run the following commands if needed:

1. If **CommandTimeOutPedSet** is missing in the Luna section run this command to add it:

```
/usr/safenet/lunaclient/bin/configurator setValue -s Luna -e CommandTimeOutPedSet -v 720000
```

2. If **PedTimeout2** value is smaller than 200000 run this command:

```
/usr/safenet/lunaclient/bin/configurator setValue -s Luna -e PEDTimeout2 -v 200000
```

To configure Windows clients

On Windows clients, already having **crystoki.ini**, any new entry provided by the newer release of the SafeNet HSM client is added to the file. But existing entry values are not modified. Manually edit the **crystoki.ini** file and modify the needed entries as follows:

1. Set **CommandTimeOutPedSet** to 720000
2. Set **PEDTimeout** to 200000.

To Switch Off Small Form-Factor Backup

If you have concerns about the physical security of your HSMs, and wish to ensure that sensitive application partition contents cannot be backed-up onto a very portable, concealable SFF token, then simply do not purchase or apply a Small Form-Factor capability update for that HSM.

If the SFF Capability Update has been installed, and for any reason you wish to disable the ability to backup HSM content, or application partition objects, to a Small Form-Factor device, you must disable HSM Policy 38.



WARNING! Disabling SFF is HSM-wide and is destructive, meaning that HSM contents and partitions are lost. Re-initialization is required, and lost objects must be re-created or must be restored from a SafeNet Backup HSM or by synchronization in an HA group.

To disable the Small Form-Factor feature

1. Enter the following command. You must be logged in as the HSM SO.

```
lunacm:>hsm changehsmpolicy -policy 38 -value 0
```

Configuration File Summary

Many aspects of SafeNet HSM configuration and operation are controlled or adjusted by the Chrystoki.conf file (Linux/UNIX) or Crystoki.ini file (Windows).

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

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

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

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

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

Where the range is a file path, <luna_client_dir> specifies the path to your SafeNet HSM client installation, for example <luna_client_dir> on Windows.

Setting	Range (Default)	Description
[Chrystoki2]		
LibNT=	(<luna_client_dir>\cryptoki.dll)	Path to the Chrystoki2 library
[Luna]		
PEDTimeout1=	(100000)	Specifies the PED timeout time 1 - defines how long the HSM tries to detect if it can talk to the PED before starting the actual communication with it. If the PED is unreachable the HSM returns to the host a result code for the respective HSM command. The result code indicates that the PED is not connected. This timeout is intended to be small so that the user is informed quickly that the PED is not connected.

Setting	Range (Default)	Description
PEDTimeout2=	(200000)	Specifies the PED timeout time 2 - defines how long the firmware waits for the local PED to respond to PED commands. PED commands should not be confused with PED-related HSM commands. An HSM sends PED commands to the PED when processing PED-related HSM commands, such as LOGIN or PED_CONNECT. One PED-related HSM command can involve many PED commands being sent by the HSM to the PED (for example, the MofN related commands). If a local PED does not respond to the PED commands within the span of PEDTimeout2 the HSM returns an appropriate result code (such as PED_TIMEOUT) for the respective PED-related HSM command. NOTE: The (default) value of 200000 is necessary to support Small Form-Factor Backup.
PEDTimeout3=	(10000)	Specifies the PED timeout time 3 - defines additional time the firmware must wait for the remote PED to respond to PED commands. That is, the actual time the firmware waits for a remote PED to respond is PEDTimeout2 + PEDTimeout3.
DefaultTimeOut=	(500000)	Sets the default timeout interval - defines how long the HSM driver in the host system waits for HSM commands to return a result code. If the result code is not returned in that time, the driver assumes that the HSM is stuck and halts it, with the DEVICE_ERROR returned to all applications that use the HSM. Most HSM commands use this timeout. Very few exceptions exist, when a command's timeout is hard-coded in the Cryptoki library, or separate timeouts are specified in the Chrystoki.conf for certain classes of HSM commands.
CommandTimeoutPedSet=	(720000)	This is such an exception to

Setting	Range (Default)	Description
		DefaultTimeout (above). It defines timeout for all PED-related HSM commands. This class of PED-related commands can take more time than the ordinary commands that subscribe to the DefaultTimeOut value. As a rule of thumb, CommandTimeOutPedSet = DefaultTimeOut + PEDTimeout1 + PEDTimeout2 + PEDTimeout3. NOTE: The (default) value of 720000 is necessary to support Small Form-Factor Backup.
KeypairGenTimeOut=	(2700000)	The amount of time the library allows for a Keypair generate operation to return a value. Due to the random component, large key sizes can take an arbitrarily long time to generate, and this setting keeps the attempts within reasonable bounds. The default is calculated as the best balance between the inconvenience of occasional very long waits and the inconvenience of restarting a keygen operation. You can change it to suit your situation.
CloningCommandTimeout=	(300000)	

[CardReader]

RemoteCommand=	0 = false (1 = true)	This setting was used when debugging older SafeNet products. For modern products it is ignored.
LunaG5Slots=	(3)	Number of SafeNet USB HSM slots reserved so that the library will check for connected devices. Can be set to zero if you have no SafeNet USB HSMs and wish to get rid of the reserved spaces in your slot list. Can be set to any number, but is effectively limited by the number of external USB devices your host can support.

[RBS]

Setting	Range (Default)	Description
HostName=	Any hostname or IP address (0.0.0.0)	
HostPort=	Any unassigned port (1792)	
ClientAuthFile=	(<luna_client_dir>\config\clientauth.dat)	
ServerCertFile=	(<luna_client_dir>\cert\server\server.pem)	
ServerPrivKeyFile=	(<luna_client_dir>\cert\server\serverkey.pem)	
ServerSSLConfigFile=	(<luna_client_dir>\openssl.cnf)	
CmdProcessor=	(<luna_client_dir>\rbs_processor2.dll)	
NetServer=	0 = false (1 = true)	

[LunaSA Client]

HtlDir=	(<luna_client_dir>\htl\)	Location of HTL-related files - dhparams certificate, htl_client, and the logs directory
SSLConfigFile=	(<luna_client_dir>\openssl.cnf)	Location of the OpenSSL configuration file.
ReceiveTimeout=	in milliseconds (20000)	Number of milliseconds before a receive timeout
TCPKeepAlive=	0 = false (1 = true)	TCPKeepAlive is a TCP stack option, available at the LunaClient, and at the SafeNet Network HSM appliance. For SafeNet purposes, it is controlled via an entry in the Chrystoki.conf /crystoki.ini file on the LunaClient, and in an equivalent file on SafeNet Network HSM. For SafeNet HSM 6.1 and newer, a fresh client software installation includes an entry "TCPKeepAlive=1" in the "LunaSA Client" section of the configuration file Chrystoki.conf (Linux/UNIX) or crystoki.ini (Windows). Config files and certificates are normally

Setting	Range (Default)	Description
		<p>preserved through an uninstall, unless you explicitly delete them.</p> <p>As such, if you update (install) LunaClient software where you previously had an older LunaClient that did not have a TCPKeepAlive entry, one is added and set to "1" (enabled), by default. In the case of update, if TCPKeepAlive is already defined in the configuration file, then your existing setting (enabled or disabled) is preserved.</p> <p>On the SafeNet Network HSM appliance, where you do not have direct access to the file system, the TCPKeepAlive= setting is controlled by the lunash:> ntls TCPKeepAlive set command.</p> <p>The settings at the appliance and the client are independent. This allows a level of assurance, in case (for example) a firewall setting blocks in one direction.</p>
NetClient=	0 = false (1 = true)	If true library will search for network slots
ServerCAFile=	(<luna_client_dir>\cert\server\CAFile.pem)	Location, on the client, of the server certificate file (set by vtl)
ClientCertFile=	(<luna_client_dir>\cert\client\ClientNameCert.pem)	Location of the Client certificate file that is uploaded to SafeNet Network HSM for NTLS. (set by vtl)
ClientPrivKeyFile=	(<luna_client_dir>\cert\client\ClientNameKey.pem)	Location of the Client private key file. (set by vtl)
ServerName00=192.20.17.200 ServerPort00=1792 ServerHtl00=0 ServerName01= ServerPort01= ServerHtl01=		<p>Entries embedded by VTL utility, when you run "vtl addServer" command. Identifies the NTLS-linked SafeNet Network HSM servers, and determines the order in which they are polled to create a slot list.</p>

[Presentation]

Setting	Range (Default)	Description
ShowUserSlots=<slot> (<serialnumber>)	Comma-delimited list of <slotnumber> (<serialnumber>), like ShowUserSlots=1(351970018022),2 (351970018021),3(351970018020),....	Sets the starting slot for the identified partition (affects only PPSO partitions). If one PPSO slot on an HSM is specified, then any that are not listed from that HSM are not displayed.
ShowAdminTokens=	yes/(no)	Admin partitions of local HSMs are visible/(not visible) in a slot listing
ShowEmptySlots=	(0)/1	When the number of partitions on an HSM is not at the limit, unused slots are shown/(not shown).
OneBaseSlotId=	(0)/1	Causes basic slot list to start at slot number 1 instead of (0).

[HAConfiguration]

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

[Misc]

ToolsDir=	(<luna_client_dir>\)	
-----------	-----------------------	--

Setting	Range (Default)	Description
PE1746Enabled=	0 = false (1 = true)	Specifies the performance target for symmetric operations based on packet sizes. For small packets, turn off this setting.
RSAPre1863KeyGen MechRemap=	(0)/1	<p>Controls what happens on newer firmware, when calls are made to specific older mechanisms that are now discouraged due to weakness. When this item is set to 0, no re-mapping is performed. When the value is set to 1, the following re-mapping occurs if the HSM firmware permits:</p> <ul style="list-style-type: none"> •PKCS Key Gen -> 186-3 Prime key gen •X9.31 Key Gen -> 186-3 Aux Prime key gen <p>(see "Mechanism Remap for FIPS Compliance " on page 1)</p>
RSAPre1863KeyGen MechRemap=	(0)/1	<p>Controls what happens on older firmware, when specific newer mechanisms are called, that are not supported on the older firmware. When this item is set to 0, no re-mapping is performed. When the value is set to 1, the following re-mapping occurs if the HSM firmware permits:</p> <ul style="list-style-type: none"> • 186-3 Prime key gen -> PKCS Key Gen • 186-3 Aux Prime key gen -> X9.31 Key Gen <p>Intended for evaluation purposes, such as with existing integrations that require newer mechanisms, before you update to firmware that actually supports the more secure mechanisms. Be careful with this setting, which makes it appear you are getting a new, secure mechanism, when really you are getting an outdated, insecure mechanism. (see "Mechanism Remap for FIPS Compliance " on page 1)</p>

Setting	Range (Default)	Description
[Secure Trusted Channel]		
ClientIdentitiesDir=	<luna_client_dir>\data\client_ identities	Specifies the directory used to store the STC client identity.
PartitionIdentitiesDir=	<luna_client_dir>\data\partition_ identities	Specifies the directory used to store the STC partition identities exported using the LunaCM stconfig partitionid export command.
ClientTokenLib=	<p>For soft token:</p> <ul style="list-style-type: none"> • <luna_client_dir>\softtoken.dll • <luna_client_dir>\win32\softtoken.dll (32-bit Windows) <p>For hard token:</p> <ul style="list-style-type: none"> • C:\Windows\System32\etoken.dll (Windows) • /usr/lib/libetoken.so (32-bit Linux/UNIX) • /usr/lib64/libetoken.so (64-bit Linux/UNIX) 	<p>Specifies the location of the token library. This value must be correct in order to use a client token.</p> <p>By default, ClientTokenLib points to the location of the soft token library. If you are using a hard token, you must manually change this value to point to the hard token library for your operating system.</p>
SoftTokenDir=	<luna_client_dir>\softtoken	<p>Specifies the location where the STC client soft token (token.db) is stored. Each client soft token is stored in its own numbered subdirectory.</p> <p>Note: In this release there is only one client token, which is stored in the 001 subdirectory.</p>

5

Domains

This chapter describes cloning domains. It contains the following sections:

- "Single Domain Policy" below
- "Legacy Domains and Migration" on the next page

Single Domain Policy

The HSM is able to support multiple partitions [See Note * below], each with its own cloning domain, if desired, as well as partition authentication for administrative users (black User PED Key for PED-authenticated HSMs, etc.) and for clients/applications (the partition challenge secret). It is possible to force all partitions on the HSM to use the same cloning domain as the administrative partition (the SO space), by setting the "Force Single Domain" policy to "Yes". This would normally be decided before any user partitions have been created on the HSM, because it is a destructive policy change, meaning that any existing HSM contents and partitions are destroyed when this policy changes. This is a convenience feature. It does not affect other authentication secrets that apply to individual partitions on the HSM.

- If the policy is set to "No" - not in force - then whenever a new partition is created, the SO is prompted to create a new cloning domain for that partition, or to imprint the partition with an existing domain. By re-using existing domain secrets, you can cause partitions to share domains, if desired, but that is optional and not forced while the policy is set to "No".
- If the policy is set to "Yes" - in force - then that prompt is skipped and each new partition is automatically assigned the cloning domain that is already in use for the HSM SO / administrative partition.
- If the policy is set to yes, then the Domain PED Key cannot have a PED PIN

Changing policies marked "destructive" will zeroize (erase completely) the entire HSM.

Description	Value	Code	Destructive
=====	=====	=====	=====
Allow masking	On	6	Yes
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
Allow offboard storage	On	22	Yes
Allow partition groups	On	23	No
Allow remote PED usage	On	25	No
Allow Acceleration	On	29	Yes
Allow unmasking	On	30	Yes
Allow FW5 compatibility mode	Off	31	No
Force Single Domain	On	35	Yes
Allow Unified PED Key	On	36	No

The HSM is NOT in FIPS 140-2 approved operation mode.

```
Command Result : 0 (Success)
[local_host] lush:>
```



Note: For SafeNet USB HSM and SafeNet PCI-E HSM, two partitions can exist, the HSM Security Officer/administrative partition (as long as the HSM has been initialized), and a single User/Application partition (once that has been created).
For SafeNet Network HSM, up to 101 partitions can exist, the HSM Security Officer/administrative partition (as long as the HSM has been initialized), and up to 100 User/Application partitions depending on purchased-or-upgraded configuration (once those are created).

Legacy Domains and Migration

The "Legacy Cloning Domain" for Password authenticated HSM partitions is the text string that was used as a cloning domain on the legacy token HSM whose contents are to be migrated to the SafeNet Network HSM partition.

The "Legacy Cloning Domain" for PED authenticated HSM partitions is the cloning domain secret on the red PED key for the legacy PED authenticated token HSM whose contents are to be migrated to the SafeNet Network HSM partition.

Your **target** SafeNet Network HSM partition has, and retains, whatever modern partition cloning domain was imprinted (on a red PED Key) when the partition was created. The "partition setLegacyDomain" command takes the domain value from your legacy HSM's red PED Key and associates that with the modern-format domain of the partition, to allow the partition to be the cloning (restore...) recipient of objects from the legacy (token) HSM.

You can repeat the "partition setLegacyDomain" command in SafeNet Shell (lunash:>) or in Lunacm, appending a different legacy domain to the partition's own domain, allowing you to consolidate the content of multiple legacy HSMs/Tokens onto a single modern partition, if desired.

The following table illustrates what happens when objects from several legacy tokens (SafeNet CA4) are migrated to SafeNet Network HSM 5 partitions. Shown are different scenarios for the legacy domain(s) and for the SafeNet Network HSM partition domain(s).

Source Token/HSM			Target HSM Partition		
Token Name	Token Contents	Token Domain	Partition Name	Partition Contents	Partition Domain

Example = four legacy tokens (different legacy domains) to four partitions (where all partitions have different modern domains)

MyToken1	Key1a, Key1b, Cert1	LegacyDomain1	MyPartition1	Key1a, Key1b, Cert1	ModemDomain1 (with LegacyDomain1 set)
MyToken2	Key2a, Key2b, Cert2	LegacyDomain2	MyPartition2	Key2a, Key2b, Cert2	ModemDomain2 (with

Source Token/HSM			Target HSM Partition		
Token Name	Token Contents	Token Domain	Partition Name	Partition Contents	Partition Domain
					LegacyDomain2 set)
MyToken3	Key3a, Key3b, Cert3	LegacyDomain3	MyPartition3	Key3a, Key3b, Cert3	ModernDomain3 (with LegacyDomain3 set)
MyToken4	Key4a, Key4b, Cert4	LegacyDomain4	MyPartition4	Key4a, Key4b, Cert4	ModernDomain4 (with LegacyDomain4 set)

Example = four legacy tokens (different legacy domains) to four partitions (where all partitions have same modern domain)

MyToken1	Key1a, Key1b, Cert1	LegacyDomain1	MyPartition1	Key1a, Key1b, Cert1	ModernDomain1 (with LegacyDomain1 set)
MyToken2	Key2a, Key2b, Cert2	LegacyDomain2	MyPartition2	Key2a, Key2b, Cert2	ModernDomain1 (with LegacyDomain2 set)
MyToken3	Key3a, Key3b, Cert3	LegacyDomain3	MyPartition3	Key3a, Key3b, Cert3	ModernDomain1 (with LegacyDomain3 set)
MyToken4	Key4a, Key4b, Cert4	LegacyDomain4	MyPartition4	Key4a, Key4b, Cert4	ModernDomain1 (with LegacyDomain4 set)

Example = four legacy tokens (shared legacy domain) to four partitions (where all partitions have different modern domains)

MyToken1	Key1a, Key1b, Cert1	Common LegacyDomain1	MyPartition1	Key1a, Key1b, Cert1	ModernDomain1 (with LegacyDomain1 set)
MyToken2	Key2a, Key2b, Cert2		MyPartition2	Key2a, Key2b, Cert2	ModernDomain2 (with LegacyDomain1 set)
MyToken3	Key3a, Key3b, Cert3		MyPartition3	Key3a, Key3b, Cert3	ModernDomain3 (with LegacyDomain1 set)
MyToken4	Key4a, Key4b, Cert4		MyPartition4	Key4a, Key4b, Cert4	ModernDomain4 (with

Source Token/HSM			Target HSM Partition		
Token Name	Token Contents	Token Domain	Partition Name	Partition Contents	Partition Domain
					LegacyDomain1 set)

Example = four legacy tokens (shared legacy domain) to four partitions (where all partitions have same modern domain)

MyToken1	Key1a, Key1b, Cert1	Common LegacyDomain1	MyPartition1	Key1a, Key1b, Cert1	ModernDomain1 (with LegacyDomain1 set i.e., same modern domain for all 4 partitions and same legacy domain associated to all 4 partitions)
MyToken2	Key2a, Key2b, Cert2		MyPartition2	Key2a, Key2b, Cert2	
MyToken3	Key3a, Key3b, Cert3		MyPartition3	Key3a, Key3b, Cert3	
MyToken4	Key4a, Key4b, Cert4		MyPartition4	Key4a, Key4b, Cert4	

Example = four legacy tokens to one partition (legacy tokens all have same domain - run "partition setLegacyDomain" once before starting to clone the first legacy token content)

MyToken1	Key1a, Key1b, Cert1	Common LegacyDomain1	MyPartition1	Key1a, Key1b, Cert1	ModernDomain1 (with LegacyDomain1 set)
MyToken2	Key2a, Key2b, Cert2			Key2a, Key2b, Cert2	
MyToken3	Key3a, Key3b, Cert3			Key3a, Key3b, Cert3	
MyToken4	Key4a, Key4b, Cert4			Key4a, Key4b, Cert4 (i.e. contents of 4 tokens into one partition)	

Example = four legacy tokens to one partition (legacy tokens all have different domains - run "partition setLegacyDomain" once before starting to clone each and EVERY legacy token's content)

MyToken1	Key1a, Key1b, Cert1	LegacyDomain1	MyPartition1	Key1a, Key1b, Cert1 Key2a, Key2b, Cert2 Key3a, Key3b, Cert3 Key4a, Key4b, Cert4 (i.e. contents of	ModernDomain1 (with LegacyDomain1 set)
----------	---------------------	---------------	--------------	---	--

Source Token/HSM			Target HSM Partition		
Token Name	Token Contents	Token Domain	Partition Name	Partition Contents	Partition Domain
MyToken2	Key2a, Key2b, Cert2	LegacyDomain2		4 tokens into one partition)	ModernDomain1 (with LegacyDomain2 set)
MyToken3	Key3a, Key3b, Cert3	LegacyDomain3			ModernDomain1 (with LegacyDomain3 set)
MyToken4	Key4a, Key4b, Cert4	LegacyDomain4			ModernDomain1 (with LegacyDomain4 set)

Contact SafeNet Technical Support -- e-mail: support@safenet-inc.com or phone 800-545-6608 (+1 410-931-7520 International) for the relevant Key Migration document, which includes explicit instructions to migrate your cryptographic objects between different types of SafeNet HSM (generally from legacy models to current models of HSM).

Error Codes and Troubleshooting

This chapter lists the HSM error codes and offers troubleshooting tips for some common issues. It contains the following sections:

- "General Troubleshooting Tips" below
- "System Operational and Error Messages" on the next page
- "Keycard and Token Return Codes " on page 174
- "Library Codes" on page 188

General Troubleshooting Tips

Here are just a few quick things to check if you are experiencing problems:

- Check the output of the syslog for any information on potential problems (`syslog tail`).
- If you see an apparent 'hang' condition, connect and check the PED - it may be waiting for a PED action.
- Check if you allowed the PED to time out, or if you started a command that needed PED action while the PED was not connected. You will need to re-issue the failed command after re-inserting the token, and pay attention to the PED.
- If RSA signing seems slow, check the Capabilities and Policies to ensure that Confirmation (policy #29) is switched off - if your security policy demands that signing operations must be verified on the HSM, then expect almost a 50% performance reduction
- If you perform a Restore from Backup operation and some or all of the objects are shown with an error message like "LUNA_RET_SM_ACCESS_DOES_NOT_VALIDATE", you might have interrupted the restore operation (even a `partition showContents` command could have this effect). Re-issue the Restore command, ensuring that no other commands are run against the partition while the operation is in progress - if other persons might be using their own ssh sessions to access the appliance, it might be best to disconnect the network cable[s] and perform your restore operation from the local (serial) console.

Remote PED

If you find that Windows fails to detect SafeNet PED, especially if you have disconnected and reconnected the PED's USB cable to your computer the PED may not be receiving adequate power. SafeNet PED is powered by PED port connection only when it is connected to a SafeNet HSM. When SafeNet PED is used for Remote PED, it is connected to a computer USB port, which does not have the same electrical characteristics as the PED port on a SafeNet HSM. The PED switches on, but might not receive sufficient power to operate.

- If you are connecting locally, always connect the PED to the SafeNet HSM.
- If you are connecting to a computer for use as a Remote PED server, always connect the PED power supply in addition to the USB connection.

System Operational and Error Messages

Why do I often see extra slots that say "token not present"?

This happens for two reasons:

- PKCS#11 originated in a world of software cryptography, which only later acknowledged the existence of Hardware Security Modules, so initially it did not have the concept of physically removable crypto slots. PKCS#11 requires a static list of slots when an application starts. The cryptographic "token" can be inserted into, or removed from a slot dynamically (by a user), for the duration of the application.
- When the token is inserted, the running application must be able to detect that token. When the token is removed, the running application gets "token not present". Because we allow for the possibility of backup, and of "PKI Bundle", we routinely declare 'place-holder' slots that might later be filled by a physical SafeNet USB HSM, or a SafeNet Backup HSM, or a SafeNet DOCK2 with (potentially) two legacy token-style HSMs in its card-reader slots. As it happens, there are three (3) USB ports on a SafeNet Network HSM appliance, so we are allowing for a physical HSM connection to all of them.

In the Chrystoki.conf file (or the Windows crystoki.ini file), for SafeNet USB HSM, you can remove the empty slots by modifying the CardReader entry, like this:

```
CardReader = {
  LunaG5Slots=0;
}
```

For SafeNet Network HSM, which has its configuration file internal to the appliance, and not directly accessible for modification, you cannot change the default cryptographic slot allotments.

Error: 'hsm update firmware' failed. (10A0B : LUNA_RET_OPERATION_RESTRICTED) when attempting to perform hsm update firmware?

You must ensure that SRK is disabled before you run the firmware update. (SRK is fundamental to Secure Transport Mode and to enforced tamper-event acknowledgement in PED-authenticated SafeNet HSMs). This brings the external split of the MTK (the Secure Recovery Vector) back inside the HSM.

Also, as with any update, you should backup any important HSM contents before proceeding.

After the update is completed, you can enable SRK again. This creates a new split of the MTK to populate a new purple PED Key.

(Applies to PED-authenticated SafeNet HSMs.)

KR_ECC_POINT_INVALID Error when decrypting a file encrypted from BSAFE through ECIES using ECC key with any of the curves from the x9_t2 section.

As indicated on the BSAFE web site, they support only the NIST-approved curves (prime, Binary, and Koblitz). That includes most/all the curves from test items 0 through 37 in ckdemo, which is to say: the "secp", "X9_62_prime", and "sect" curves.

The X9.62 curves that are failing in this task are X9.62 binary/char2 curves which do not appear to be supported by BSAFE. So, you appear to be encountering a BSAFE limitation and not a SafeNet HSM problem.

Slow/interrupted response from the HSM, and the "hsm show" command shows LUNA_RET_SM_SESSION_REALLOC_ERROR

```
Appliance Details:
=====
Software Version:          4.4.0-27
Error: 'hsm show' failed. (310102 : LUNA_RET_SM_SESSION_REALLOC_ERROR)
```

Command Result : 65535 (Luna Shell execution)

The error LUNA_RET_SM_SESSION_REALLOC_ERROR means the HSM cannot expand the session table.

The HSM maintains a table for all of the open sessions. For performance reasons, the table is quite small initially. As sessions are opened (and not closed) the table fills up. When the table gets full, the HSM tries to expand the table. If there is not enough available RAM to grow the table, this error is returned.

RAM can be used up by an application that creates **and does not delete** a large number of session objects, as well as by an application that opens and **fails to close** a large number of sessions.

The obvious solution is proper housekeeping. Your applications **MUST** clean up after themselves, by closing sessions that are no longer in use - this deletes session objects associated with those sessions. If your application practice is to have long-lived sessions, and to open many objects in a given session, then your application should explicitly delete those session objects as soon as each one is no longer necessary.

By far, we see more of the former problem - abandoned sessions - and very often in conjunction with Java-based applications. Proper garbage collection includes deleting session objects when they are no longer useful, or simply closing sessions as soon as they are not required. Formally closing a session (or stopping / restarting the HSM) deletes all session objects within each affected session. These actions keep the session table small, so it uses the least possible HSM volatile memory.

Low Battery Message

The K6 HSM card, used in the SafeNet Network HSM and SafeNet PCI-E HSM products, is equipped with a non-replaceable battery that is expected to last the life of the product. If you notice a log message or other warning about 'battery low', or similar, contact SafeNet Technical Support.

Keycard and Token Return Codes

The following table summarizes HSM error codes (last updated for firmware 6.10.1) :

HSM Error	Hex Code	PKCS#11 or SFNT Defined CKR Error
LUNA_RET_OK	0x00000000	CKR_OK
LUNA_RET_CANCEL	0x00010000	CKR_CANCEL
LUNA_RET_FLAGS_INVALID	0x00040000	CKR_FLAGS_INVALID, removed from v2.0
LUNA_RET_TOKEN_NOT_PRESENT	0x00E00000	CKR_TOKEN_NOT_PRESENT
LUNA_RET_FORMER_INVALID_ENTRY_TYPE	0x00300130	CKR_DEVICE_ERROR
LUNA_RET_SP_TX_ERROR	0x00300131	CKR_DEVICE_ERROR

HSM Error	Hex Code	PKCS#11 or SFNT Defined CKR Error
LUNA_RET_SP_RX_ERROR	0x00300132	CKR_DEVICE_ERROR
LUNA_RET_PED_ID_INVALID	0x00300140	CKR_DEVICE_ERROR
LUNA_RET_PED_UNSUPPORTED_PROTOCOL	0x00300141	CKR_DEVICE_ERROR
LUNA_RET_PED_UNPLUGGED	0x00300142	CKR_PED_UNPLUGGED
LUNA_RET_PED_ERROR	0x00300144	CKR_DEVICE_ERROR
LUNA_RET_PED_UNSUPPORTED_CRYPTO_PROTOCOL	0x00300145	CKR_DEVICE_ERROR
LUNA_RET_PED_DEK_INVALID	0x00300146	CKR_DEVICE_ERROR
LUNA_RET_PED_CLIENT_NOT_RUNNING	0x00300147	CKR_PED_CLIENT_NOT_RUNNING
LUNA_RET_CL_ALIGNMENT_ERROR	0x00300200	CKR_DEVICE_ERROR
LUNA_RET_CL_QUEUE_LOCATION_ERROR	0x00300201	CKR_DEVICE_ERROR
LUNA_RET_CL_QUEUE_OVERLAP_ERROR	0x00300202	CKR_DEVICE_ERROR
LUNA_RET_CL_TRANSMISSION_ERROR	0x00300203	CKR_DEVICE_ERROR
LUNA_RET_CL_NO_TRANSMISSION	0x00300204	CKR_DEVICE_ERROR
LUNA_RET_CL_COMMAND_MALFORMED	0x00300205	CKR_DEVICE_ERROR
LUNA_RET_CL_MAILBOXES_NOT_AVAILABLE	0x00300206	CKR_DEVICE_ERROR
LUNA_RET_MM_NOT_ENOUGH_MEMORY	0x00310000	CKR_DEVICE_ERROR
LUNA_RET_MM_INVALID_HANDLE	0x00310001	CKR_DEVICE_ERROR
LUNA_RET_MM_USAGE_ALREADY_SET	0x00310002	CKR_DEVICE_ERROR
LUNA_RET_MM_ACCESS_OUTSIDE_ALLOCATION_RANGE	0x00310003	CKR_DEVICE_ERROR
LUNA_RET_MM_INVALID_USAGE	0x00310004	CKR_DEVICE_ERROR
LUNA_RET_MM_ITERATOR_PAST_END	0x00310005	CKR_DEVICE_ERROR
LUNA_RET_MM_FATAL_ERROR	0x00310006	CKR_DEVICE_ERROR
LUNA_RET_TEMPLATE_INCOMPLETE	0x00D00000	CKR_TEMPLATE_INCOMPLETE
LUNA_RET_TEMPLATE_INCONSISTENT	0x00D10000	CKR_TEMPLATE_INCONSISTENT *
LUNA_RET_ATTRIBUTE_TYPE_INVALID	0x00120000	CKR_ATTRIBUTE_TYPE_INVALID
LUNA_RET_ATTRIBUTE_VALUE_INVALID	0x00130000	CKR_ATTRIBUTE_VALUE_INVALID

HSM Error	Hex Code	PKCS#11 or SFNT Defined CKR Error
LUNA_RET_ATTRIBUTE_READ_ONLY	0x00100000	CKR_ATTRIBUTE_READ_ONLY
LUNA_RET_ATTRIBUTE_SENSITIVE	0x00110000	CKR_ATTRIBUTE_SENSITIVE
LUNA_RET_OBJECT_HANDLE_INVALID	0x00820000	CKR_OBJECT_HANDLE_INVALID
LUNA_RET_MAX_OBJECT_COUNT	0x00820001	CKR_MAX_OBJECT_COUNT_EXCEEDED
LUNA_RET_ATTRIBUTE_NOT_FOUND	0x00120010	CKR_ATTRIBUTE_TYPE_INVALID
LUNA_RET_CAN_NOT_CREATE_SECRET_KEY	0x00D10011	CKR_TEMPLATE_INCONSISTENT
LUNA_RET_CAN_NOT_CREATE_PRIVATE_KEY	0x00D10012	CKR_TEMPLATE_INCONSISTENT
LUNA_RET_SECRET_KEY_MUST_BE_SENSITIVE	0x00130013	CKR_ATTRIBUTE_VALUE_INVALID
LUNA_RET_SECRET_KEY_MUST_HAVE_SENSITIVE_ATTRIBUTE	0x00D00014	CKR_TEMPLATE_INCOMPLETE
LUNA_RET_PRIVATE_KEY_MUST_BE_SENSITIVE	0x00130015	CKR_ATTRIBUTE_VALUE_INVALID
LUNA_RET_PRIVATE_KEY_MUST_HAVE_SENSITIVE_ATTRIBUTE	0x00D00016	CKR_TEMPLATE_INCOMPLETE
LUNA_RET_SIGNING_KEY_MUST_BE_LOCAL	0x00680001	CKR_KEY_FUNCTION_NOT_PERMITTED
LUNA_RET_MULTI_FUNCTION_KEYS_NOT_ALLOWED	0x00D10018	CKR_TEMPLATE_INCONSISTENT
LUNA_RET_CAN_NOT_CHANGE_KEY_FUNCTION	0x00100019	CKR_ATTRIBUTE_READ_ONLY
LUNA_RET_KEY_SIZE_RANGE	0x00620000	CKR_KEY_SIZE_RANGE
LUNA_RET_KEY_TYPE_INCONSISTENT	0x00630000	CKR_KEY_TYPE_INCONSISTENT
LUNA_RET_KEY_INVALID_FOR_OPERATION	0x00630001	CKR_KEY_TYPE_INCONSISTENT
LUNA_RET_KEY_PARITY	0x00630002	CKR_KEY_TYPE_INCONSISTENT
LUNA_RET_KEY_UNEXTRACTABLE	0x006a0000	CKR_KEY_UNEXTRACTABLE
LUNA_RET_KEY_EXTRACTABLE	0x006a0001	CKR_KEY_UNEXTRACTABLE
LUNA_RET_KEY_INDIGESTIBLE	0x00670000	CKR_KEY_INDIGESTIBLE
LUNA_RET_KEY_NOT_WRAPPABLE	0x00690000	CKR_KEY_NOT_WRAPPABLE
LUNA_RET_KEY_NOT_UNWRAPPABLE	0x00690001	CKR_KEY_NOT_WRAPPABLE

HSM Error	Hex Code	PKCS#11 or SFNT Defined CKR Error
LUNA_RET_ARGUMENTS_BAD	0x00070000	CKR_ARGUMENTS_BAD
LUNA_RET_INVALID_ENTRY_TYPE	0x00070001	CKR_INVALID_ENTRY_TYPE
LUNA_RET_DATA_INVALID	0x00200000	CKR_DATA_INVALID
LUNA_RET_SM_DATA_INVALID	0x00200002	CKR_DATA_INVALID
LUNA_RET_NO_RNG_SEED	0x00200015	CKR_DATA_INVALID
LUNA_RET_FUNCTION_NOT_SUPPORTED	0x00540000	CKR_FUNCTION_NOT_SUPPORTED
LUNA_RET_NO_OFFBOARD_STORAGE	0x00540001	CKR_FUNCTION_NOT_SUPPORTED
LUNA_RET_CL_COMMAND_NON_BACKUP	0x00540002	CKR_FUNCTION_NOT_SUPPORTED
LUNA_RET_BUFFER_TOO_SMALL	0x01500000	CKR_BUFFER_TOO_SMALL
LUNA_RET_DATA_LEN_RANGE	0x00210000	CKR_DATA_LEN_RANGE
LUNA_RET_GENERAL_ERROR	0x00050000	CKR_GENERAL_ERROR
LUNA_RET_DEVICE_ERROR	0x00300000	CKR_DEVICE_ERROR
LUNA_RET_UNKNOWN_COMMAND	0x00300001	CKR_FUNCTION_NOT_SUPPORTED
LUNA_RET_TOKEN_LOCKED_OUT	0x00300002	CKR_PIN_LOCKED
LUNA_RET_RNG_ERROR	0x00300003	CKR_DEVICE_ERROR
LUNA_RET_DES_SELF_TEST_FAILURE	0x00300004	CKR_DEVICE_ERROR
LUNA_RET_CAST_SELF_TEST_FAILURE	0x00300005	CKR_DEVICE_ERROR
LUNA_RET_CAST3_SELF_TEST_FAILURE	0x00300006	CKR_DEVICE_ERROR
LUNA_RET_CAST5_SELF_TEST_FAILURE	0x00300007	CKR_DEVICE_ERROR
LUNA_RET_MD2_SELF_TEST_FAILURE	0x00300008	CKR_DEVICE_ERROR
LUNA_RET_MD5_SELF_TEST_FAILURE	0x00300009	CKR_DEVICE_ERROR
LUNA_RET_SHA_SELF_TEST_FAILURE	0x0030000a	CKR_DEVICE_ERROR
LUNA_RET_RSA_SELF_TEST_FAILURE	0x0030000b	CKR_DEVICE_ERROR
LUNA_RET_RC2_SELF_TEST_FAILURE	0x0030000c	CKR_DEVICE_ERROR
LUNA_RET_RC4_SELF_TEST_FAILURE	0x0030000d	CKR_DEVICE_ERROR
LUNA_RET_RC5_SELF_TEST_FAILURE	0x0030000e	CKR_DEVICE_ERROR
LUNA_RET_SO_LOGIN_FAILURE_THRESHOLD	0x0030000f	CKR_SO_LOGIN_FAILURE_THRESHOLD

HSM Error	Hex Code	PKCS#11 or SFNT Defined CKR Error
LUNA_RET_RNG_SELF_TEST_FAILURE	0x00300010	CKR_DEVICE_ERROR
LUNA_RET_SM_UNKNOWN_COMMAND	0x00300011	CKR_DEVICE_ERROR
LUNA_RET_UM_TSN_MISSING	0x00300012	CKR_DEVICE_ERROR
LUNA_RET_SM_TSV_MISSING	0x00300013	CKR_DEVICE_ERROR
LUNA_RET_SM_UNKNOWN_TOSM_STATE	0x00300014	CKR_DEVICE_ERROR
LUNA_RET_DSA_PARAM_GEN_FAILURE	0x00300015	CKR_DEVICE_ERROR
LUNA_RET_DSA_SELF_TEST_FAILURE	0x00300016	CKR_DEVICE_ERROR
LUNA_RET_SEED_SELF_TEST_FAILURE	0x00300017	CKR_DEVICE_ERROR
LUNA_RET_AES_SELF_TEST_FAILURE	0x00300018	CKR_DEVICE_ERROR
LUNA_RET_FUNCTION_NOT_SUPPORTED_BY_HARDWARE	0x00300019	CKR_DEVICE_ERROR
LUNA_RET_HAS160_SELF_TEST_FAILURE	0x0030001a	CKR_DEVICE_ERROR
LUNA_RET_KCDSA_PARAM_GEN_FAILURE	0x0030001b	CKR_DEVICE_ERROR
LUNA_RET_KCDSA_SELF_TEST_FAILURE	0x0030001c	CKR_DEVICE_ERROR
LUNA_RET_HSM_INTERNAL_BUFFER_TOO_SMALL	0x0030001d	CKR_DEVICE_ERROR
LUNA_RET_COUNTER_WRAPAROUND	0x0030001e	CKR_DEVICE_ERROR
LUNA_RET_TIMEOUT	0x0030001f	CKR_TIMEOUT
LUNA_RET_NOT_READY	0x00300020	CKR_DEVICE_ERROR
LUNA_RET_RETRY	0x00300021	CKR_DEVICE_ERROR
LUNA_RET_SHA1_RSA_SELF_TEST_FAILURE	0x00300022	CKR_DEVICE_ERROR
LUNA_RET_SELF_TEST_FAILURE	0x00300023	CKR_DEVICE_ERROR
LUNA_RET_INCOMPATIBLE	0x00300024	CKR_DEVICE_ERROR
LUNA_RET_RIPEMD160_SELF_TEST_FAILURE	0x00300034	CKR_DEVICE_ERROR
LUNA_RET_TOKEN_LOCKED_OUT_CL	0x00300100	CKR_DEVICE_ERROR
LUNA_RET_TOKEN_LOCKED_OUT_MM	0x00300101	CKR_DEVICE_ERROR
LUNA_RET_TOKEN_LOCKED_OUT_UM	0x00300102	CKR_DEVICE_ERROR
LUNA_RET_TOKEN_LOCKED_OUT_SM	0x00300103	CKR_DEVICE_ERROR
LUNA_RET_TOKEN_LOCKED_OUT_RN	0x00300104	CKR_DEVICE_ERROR

HSM Error	Hex Code	PKCS#11 or SFNT Defined CKR Error
LUNA_RET_TOKEN_LOCKED_OUT_CA	0x00300105	CKR_DEVICE_ERROR
LUNA_RET_TOKEN_LOCKED_OUT_PM	0x00300106	CKR_DEVICE_ERROR
LUNA_RET_TOKEN_LOCKED_OUT_OH	0x00300107	CKR_DEVICE_ERROR
LUNA_RET_TOKEN_LOCKED_OUT_CCM	0x00300108	CKR_DEVICE_ERROR
LUNA_RET_TOKEN_LOCKED_OUT_SHA_DIGEST	0x00300109	CKR_DEVICE_ERROR
LUNA_RET_SM_ACCESS_REALLOC_ERROR	0x00310101	CKR_DEVICE_ERROR
LUNA_RET_SM_SESSION_REALLOC_ERROR	0x00310102	CKR_DEVICE_ERROR
LUNA_RET_SM_MEMORY_ALLOCATION_ERROR	0x00310103	CKR_DEVICE_ERROR
LUNA_RET_ENCRYPTED_DATA_INVALID	0x00400000	CKR_ENCRYPTED_DATA_INVALID
LUNA_RET_ENCRYPTED_DATA_LEN_RANGE	0x00410000	CKR_ENCRYPTED_DATA_LEN_RANGE
LUNA_RET_FUNCTION_CANCELED	0x00500000	CKR_FUNCTION_CANCELED
LUNA_RET_KEY_HANDLE_INVALID	0x00600000	CKR_KEY_HANDLE_INVALID
LUNA_RET_MECHANISM_INVALID	0x00700000	CKR_MECHANISM_INVALID
LUNA_RET_MECHANISM_PARAM_INVALID	0x00710000	CKR_MECHANISM_PARAM_INVALID
LUNA_RET_OPERATION_ACTIVE	0x00900000	CKR_OPERATION_ACTIVE
LUNA_RET_OPERATION_NOT_INITIALIZED	0x00910000	CKR_OPERATION_NOT_INITIALIZED
LUNA_RET_UM_PIN_INCORRECT	0x00a00000	CKR_PIN_INCORRECT
LUNA_RET_UM_PIN_INCORRECT_CONTAINER_ZEROIZED	0x00a00001	CKR_PIN_INCORRECT
LUNA_RET_UM_PIN_INCORRECT_CONTAINER_LOCKED	0x00a00002	CKR_PIN_INCORRECT
LUNA_RET_UM_PIN_LEN_RANGE	0x00a20000	CKR_PIN_LEN_RANGE
LUNA_RET_SM_PIN_EXPIRED	0x00a30000	CKR_PIN_EXPIRED
LUNA_RET_SM_EXCLUSIVE_SESSION_EXISTS	0x00b20000	CKR_SESSION_EXCLUSIVE_EXISTS
LUNA_RET_SM_SESSION_HANDLE_INVALID	0x00b30000	CKR_SESSION_HANDLE_INVALID
LUNA_RET_SIGNATURE_INVALID	0x00c00000	CKR_SIGNATURE_INVALID

HSM Error	Hex Code	PKCS#11 or SFNT Defined CKR Error
LUNA_RET_SIGNATURE_LEN_RANGE	0x00c10000	CKR_SIGNATURE_LEN_RANGE
LUNA_RET_UNWRAPPING_KEY_HANDLE_INVALID	0x00f00000	CKR_UNWRAPPING_KEY_HANDLE_INVALID
LUNA_RET_UNWRAPPING_KEY_SIZE_RANGE	0x00f10000	CKR_UNWRAPPING_KEY_SIZE_RANGE
LUNA_RET_UNWRAPPING_KEY_TYPE_INCONSISTENT	0x00f20000	CKR_UNWRAPPING_KEY_TYPE_INCONSISTENT
LUNA_RET_USER_ALREADY_LOGGED_IN	0x01000000	CKR_USER_ALREADY_LOGGED_IN
LUNA_RET_SM_OTHER_USER_LOGGED_IN	0x01000001	CKR_USER_ALREADY_LOGGED_IN
LUNA_RET_USER_NOT_LOGGED_IN	0x01010000	CKR_USER_NOT_LOGGED_IN
LUNA_RET_SM_NOT_LOGGED_IN	0x01010001	CKR_USER_NOT_LOGGED_IN
LUNA_RET_USER_PIN_NOT_INITIALIZED	0x01020000	CKR_USER_PIN_NOT_INITIALIZED
LUNA_RET_USER_TYPE_INVALID	0x01030000	CKR_USER_TYPE_INVALID
LUNA_RET_WRAPPED_KEY_INVALID	0x01100000	CKR_WRAPPED_KEY_INVALID
LUNA_RET_WRAPPED_KEY_LEN_RANGE	0x01120000	CKR_WRAPPED_KEY_LEN_RANGE
LUNA_RET_WRAPPING_KEY_HANDLE_INVALID	0x01130000	CKR_WRAPPING_KEY_HANDLE_INVALID
LUNA_RET_WRAPPING_KEY_SIZE_RANGE	0x01140000	CKR_WRAPPING_KEY_SIZE_RANGE
LUNA_RET_WRAPPING_KEY_TYPE_INCONSISTENT	0x01150000	CKR_WRAPPING_KEY_TYPE_INCONSISTENT
LUNA_RET_CERT_VERSION_NOT_SUPPORTED	0x00300300	CKR_DEVICE_ERROR
LUNA_RET_SIM_AUTHFORM_INVALID	0x0020011e	CKR_SIM_AUTHFORM_INVALID
LUNA_RET_CCM_TOO_LARGE	0x00210001	CKR_DATA_LEN_RANGE
LUNA_RET_TEST_VS_BSAFE_FAILED	0x00300820	CKR_DEVICE_ERROR
LUNA_RET_SFNT3120_ERROR	0x00300821	CKR_DEVICE_ERROR
LUNA_RET_SFNT3120_SELFTEST_FAILED	0x00300822	CKR_DEVICE_ERROR
LUNA_RET_SFNT3120_CRC	0x00300823	CKR_DEVICE_ERROR
LUNA_RET_SFNT3120_ALG_NO_SOFTWARE_SUPPORT	0x00300824	CKR_DEVICE_ERROR

HSM Error	Hex Code	PKCS#11 or SFNT Defined CKR Error
LUNA_RET_ISES_ERROR	0x00300880	CKR_DEVICE_ERROR
LUNA_RET_ISES_INIT_FAILED	0x00300881	CKR_DEVICE_ERROR
LUNA_RET_ISES_LNAU_TEST_FAILED	0x00300882	CKR_DEVICE_ERROR
LUNA_RET_ISES_RNG_TEST_FAILED	0x00300883	CKR_DEVICE_ERROR
LUNA_RET_ISES_CMD_FAILED	0x00300884	CKR_DEVICE_ERROR
LUNA_RET_ISES_CMD_PARAMETER_INVALID	0x00300885	CKR_DEVICE_ERROR
LUNA_RET_ISES_TEST_VS_BSAFE_FAILED	0x00300886	CKR_DEVICE_ERROR
LUNA_RET_PE1746_ERROR	0x00300887	CKR_DEVICE_ERROR
LUNA_RET_RM_ELEMENT_VALUE_INVALID	0x00200a00	CKR_DATA_INVALID
LUNA_RET_RM_ELEMENT_ID_INVALID	0x00200a01	CKR_DATA_INVALID
LUNA_RET_RM_NO_MEMORY	0x00310a02	CKR_DEVICE_MEMORY
LUNA_RET_RM_BAD_HSM_PARAMS	0x00300a03	CKR_DEVICE_ERROR
LUNA_RET_RM_POLICY_ELEMENT_DESTRUCTIVE	0x00200a04	CKR_DATA_INVALID
LUNA_RET_RM_POLICY_ELEMENT_NOT_DESTRUCTIVE	0x00200a05	CKR_DATA_INVALID
LUNA_RET_RM_CONFIG_CHANGE_ILLEGAL	0x00010a06	CKR_CANCEL
LUNA_RET_RM_CONFIG_CHANGE_FAILS_DEPENDENCIES	0x00010a07	CKR_CANCEL
LUNA_RET_LICENSE_ID_UNKNOWN	0x00200a08	CKR_DATA_INVALID
LUNA_RET_LICENSE_CAPACITY_EXCEEDED	0x00010a09	CKR_LICENSE_CAPACITY_EXCEEDED
LUNA_RET_RM_POLICY_WRITE_RESTRICTED	0x00010a0a	CKR_CANCEL
LUNA_RET_OPERATION_RESTRICTED	0x00010a0b	CKR_OPERATION_NOT_ALLOWED
LUNA_RET_CANNOT_PERFORM_OPERATION_TWICE	0x00010a0c	CKR_CANCEL
LUNA_RET_BAD_PPID	0x00200a0d	CKR_DATA_INVALID
LUNA_RET_BAD_FW_VERSION	0x00200a0e	CKR_DATA_INVALID
LUNA_RET_OPERATION_SHOULD_BE_DESTRUCTIVE	0x00200a0f	CKR_DATA_INVALID

HSM Error	Hex Code	PKCS#11 or SFNT Defined CKR Error
LUNA_RET_RM_CONFIG_ILLEGAL	0x00200a10	CKR_DATA_INVALID
LUNA_RET_BAD_SN	0x00200a11	CKR_DATA_INVALID
LUNA_RET_CHALLENGE_TYPE_INVALID	0x00200b00	CKR_DATA_INVALID
LUNA_RET_CHALLENGE_REQUIRES_PED	0x00010b01	CKR_CANCEL
LUNA_RET_CHALLENGE_NOT_REQUIRED	0x00010b02	CKR_CANCEL
LUNA_RET_CHALLENGE_RESPONSE_INCORRECT	0x00a00b03	CKR_PIN_INCORRECT
LUNA_RET_OH_OBJECT_VERSION_INVALID	0x00300c00	CKR_DEVICE_ERROR
LUNA_RET_OH_OBJECT_TYPE_INVALID	0x00300c01	CKR_DEVICE_ERROR
LUNA_RET_OH_OBJECT_ALREADY_EXISTS	0x00010c02	CKR_CANCEL
LUNA_RET_OH_OBJECT_OWNER_DOES_NOT_EXIST	0x00200c03	CKR_DATA_INVALID
LUNA_RET_STORAGE_TYPE_INCONSISTENT	0x00200c04	CKR_DATA_INVALID
LUNA_RET_CONTAINER_CAN_NOT_HAVE_MEMBERS	0x00200c05	CKR_DATA_INVALID
LUNA_RET_SAVED_STATE_INVALID	0x01600000	CKR_SAVED_STATE_INVALID
LUNA_RET_STATE_UNSAVEABLE	0x01800000	CKR_STATE_UNSAVEABLE
LUNA_RET_ERROR	0x80000000	CKR_GENERAL_ERROR
LUNA_RET_CONTAINER_HANDLE_INVALID	0x80000001	CKR_CONTAINER_HANDLE_INVALID
LUNA_RET_INVALID_PADDING_TYPE	0x80000002	CKR_DATA_INVALID
LUNA_RET_NOT_FOUND	0x80000007	CKR_FUNCTION_FAILED
LUNA_RET_TOO_MANY_CONTAINERS	0x80000008	CKR_TOO_MANY_CONTAINERS
LUNA_RET_CONTAINER_LOCKED	0x80000009	CKR_PIN_LOCKED
LUNA_RET_CONTAINER_IS_DISABLED	0x8000000a	CKR_PARTITION_DISABLED
LUNA_RET_SECURITY_PARAMETER_MISSING	0x8000000b	CKR_SECURITY_PARAMETER_MISSING
LUNA_RET_DEVICE_TIMEOUT	0x8000000c	CKR_DEVICE_TIMEOUT
LUNA_RET_OBJECT_DELETED	0x8000000d	HSM Internal ONLY
LUNA_RET_INVALID_FUF_TARGET	0x8000000e	CKR_INVALID_FUF_TARGET

HSM Error	Hex Code	PKCS#11 or SFNT Defined CKR Error
LUNA_RET_INVALID_FUF_HEADER	0x8000000f	CKR_INVALID_FUF_HEADER
LUNA_RET_INVALID_FUF_VERSION	0x80000010	CKR_INVALID_FUF_VERSION
LUNA_RET_KCV_PARAMETER_ALREADY_EXISTS	0x80000100	CKR_CLONING_PARAMETER_ALREADY_EXISTS
LUNA_RET_KCV_PARAMETER_COULD_NOT_BE_ADDED	0x80000101	CKR_DEVICE_MEMORY
LUNA_RET_INVALID_CERTIFICATE_DATA	0x80000102	CKR_CERTIFICATE_DATA_INVALID
LUNA_RET_INVALID_CERTIFICATE_TYPE	0x80000103	CKR_CERTIFICATE_DATA_INVALID
LUNA_RET_INVALID_CERTIFICATE_VERSION	0x80000104	CKR_CERTIFICATE_DATA_INVALID
LUNA_RET_INVALID_MODULUS_SIZE	0x80000105	CKR_ATTRIBUTE_VALUE_INVALID
LUNA_RET_WRAPPING_ERROR	0x80000107	CKR_WRAPPING_ERROR
LUNA_RET_UNWRAPPING_ERROR	0x80000108	CKR_UNWRAPPING_ERROR
LUNA_RET_INVALID_PRIVATE_KEY_TYPE	0x80000109	CKR_DATA_INVALID
LUNA_RET_TSN_MISMATCH	0x8000010a	CKR_DATA_INVALID
LUNA_RET_KCV_PARAMETER_MISSING	0x8000010b	CKR_CLONING_PARAMETER_MISSING
LUNA_RET_TWC_PARAMETER_MISSING	0x8000010c	CKR_CERTIFICATE_DATA_MISSING
LUNA_RET_TUK_PARAMETER_MISSING	0x8000010d	CKR_CERTIFICATE_DATA_MISSING
LUNA_RET_CPK_PARAMETER_MISSING	0x8000010e	CKR_KEY_NEEDED
LUNA_RET_MASKING_NOT_SUPPORTED	0x8000010f	CKR_FUNCTION_NOT_SUPPORTED
LUNA_RET_INVALID_ACCESS_LEVEL	0x80000110	CKR_ARGUMENTS_BAD
LUNA_RET_MAC_MISSING	0x80000111	CKR_MAC_MISSING
LUNA_RET_DAC_POLICY_PID_MISMATCH	0x80000112	CKR_DAC_POLICY_PID_MISMATCH
LUNA_RET_DAC_MISSING	0x80000113	CKR_DAC_MISSING
LUNA_RET_BAD_DAC	0x80000114	CKR_BAD_DAC
LUNA_RET_SSK_MISSING	0x80000115	CKR_SSK_MISSING
LUNA_RET_BAD_MAC	0x80000116	CKR_BAD_MAC
LUNA_RET_DAK_MISSING	0x80000117	CKR_DAK_MISSING
LUNA_RET_BAD_DAK	0x80000118	CKR_BAD_DAK

HSM Error	Hex Code	PKCS#11 or SFNT Defined CKR Error
LUNA_RET_HOK_MISSING	0x80000119	CKR_CERTIFICATE_DATA_MISSING
LUNA_RET_CITS_DAK_MISSING	0x8000011a	CKR_CITS_DAK_MISSING
LUNA_RET_SIM_AUTHORIZATION_FAILED	0x8000011b	CKR_SIM_AUTHORIZATION_FAILED
LUNA_RET_SIM_VERSION_UNSUPPORTED	0x8000011c	CKR_SIM_VERSION_UNSUPPORTED
LUNA_RET_SIM_CORRUPT_DATA	0x8000011d	CKR_SIM_CORRUPT_DATA
LUNA_RET_ECC_MIC_MISSING	0x8000011e	CKR_CERTIFICATE_DATA_MISSING
LUNA_RET_ECC_HOK_MISSING	0x8000011f	CKR_CERTIFICATE_DATA_MISSING
LUNA_RET_ECC_HOC_MISSING	0x80000120	CKR_CERTIFICATE_DATA_MISSING
LUNA_RET_ECC_DAK_MISSING	0x80000121	CKR_CERTIFICATE_DATA_MISSING
LUNA_RET_ECC_DAC_MISSING	0x80000122	CKR_CERTIFICATE_DATA_MISSING
LUNA_RET_ROOT_CERT_MISSING	0x80000123	CKR_CERTIFICATE_DATA_MISSING
LUNA_RET_HOC_MISSING	0x80000124	CKR_CERTIFICATE_DATA_MISSING
LUNA_RET_INVALID_CERTIFICATE_FUNCTION	0x80000125	CKR_CERTIFICATE_DATA_INVALID
LUNA_RET_N_TOO_LARGE	0x80000200	CKR_ARGUMENTS_BAD
LUNA_RET_N_TOO_SMALL	0x80000201	CKR_ARGUMENTS_BAD
LUNA_RET_M_TOO_LARGE	0x80000202	CKR_ARGUMENTS_BAD
LUNA_RET_M_TOO_SMALL	0x80000203	CKR_ARGUMENTS_BAD
LUNA_RET_WEIGHT_TOO_LARGE	0x80000204	CKR_ARGUMENTS_BAD
LUNA_RET_WEIGHT_TOO_SMALL	0x80000205	CKR_ARGUMENTS_BAD
LUNA_RET_TOTAL_WEIGHT_INVALID	0x80000206	CKR_ARGUMENTS_BAD
LUNA_RET_MISSING_SPLITS	0x80000207	CKR_ARGUMENTS_BAD
LUNA_RET_SPLIT_DATA_INVALID	0x80000208	CKR_ARGUMENTS_BAD
LUNA_RET_SPLIT_ID_INVALID	0x80000209	CKR_ARGUMENTS_BAD
LUNA_RET_M_OF_N_PARAMETER_NOT_AVAILABLE	0x8000020a	CKR_OPERATION_NOT_INITIALIZED
LUNA_RET_M_OF_N_ACTIVATION_REQUIRED	0x8000020b	CKR_OPERATION_NOT_INITIALIZED
LUNA_RET_TOO_MANY_WEIGHTS	0x8000020e	CKR_ARGUMENTS_BAD
LUNA_RET_MISSING_WEIGHT_VALUE	0x8000020f	CKR_ARGUMENTS_BAD

HSM Error	Hex Code	PKCS#11 or SFNT Defined CKR Error
LUNA_RET_MISSING_VALUE_FOR_M	0x80000210	CKR_ARGUMENTS_BAD
LUNA_RET_MISSING_VALUE_FOR_N	0x80000211	CKR_ARGUMENTS_BAD
LUNA_RET_MISSING_NUMBER_OF_VECTORS	0x80000212	CKR_ARGUMENTS_BAD
LUNA_RET_MISSING_VECTOR	0x80000213	CKR_ARGUMENTS_BAD
LUNA_RET_VECTOR_TOO_LARGE	0x80000214	CKR_ARGUMENTS_BAD
LUNA_RET_VECTOR_TOO_SMALL	0x80000215	CKR_ARGUMENTS_BAD
LUNA_RET_TOO_MANY_VECTORS_PROVIDED	0x80000216	CKR_ARGUMENTS_BAD
LUNA_RET_INVALID_VECTOR_SIZE	0x80000217	CKR_ARGUMENTS_BAD
LUNA_RET_M_OF_N_PARAMETER_EXIST	0x80000218	CKR_FUNCTION_FAILED
LUNA_RET_VECTOR_VERSION_INVALID	0x80000219	CKR_DATA_INVALID
LUNA_RET_VECTOR_OF_DIFFERENT_SET	0x8000021a	CKR_ARGUMENTS_BAD
LUNA_RET_VECTOR_DUPLICATE	0x8000021b	CKR_ARGUMENTS_BAD
LUNA_RET_VECTOR_TYPE_INVALID	0x8000021c	CKR_ARGUMENTS_BAD
LUNA_RET_MISSING_COMMAND_PARAMETER	0x8000021d	CKR_ARGUMENTS_BAD
LUNA_RET_M_OF_N_CLONING_IS_NOT_ALLOWED	0x8000021e	CKR_FUNCTION_NOT_SUPPORTED
LUNA_RET_M_OF_N_IS_NOT_REQUIRED	0x8000021f	CKR_OPERATION_NOT_INITIALIZED
LUNA_RET_M_OF_N_IS_NOT_INITIALIZED	0x80000220	CKR_OPERATION_NOT_INITIALIZED
LUNA_RET_M_OF_N_SECRET_INVALID	0x80000221	CKR_GENERAL_ERROR
LUNA_RET_CCM_NOT_PRESENT	0x80000300	CKR_FUNCTION_NOT_SUPPORTED
LUNA_RET_CCM_NOT_SUPPORTED	0x80000301	CKR_FUNCTION_NOT_SUPPORTED
LUNA_RET_CCM_UNREMOVABLE	0x80000302	CKR_DATA_INVALID
LUNA_RET_CCM_CERT_INVALID	0x80000303	CKR_DATA_INVALID
LUNA_RET_CCM_SIGN_INVALID	0x80000304	CKR_DATA_INVALID
LUNA_RET_CCM_UPDATE_DENIED	0x80000305	CKR_DATA_INVALID
LUNA_RET_CCM_FWUPDATE_DENIED	0x80000306	CKR_DATA_INVALID
LUNA_RET_SM_ACCESS_ID_INVALID	0x80000400	CKR_DATA_INVALID
LUNA_RET_SM_ACCESS_ALREADY_EXISTS	0x80000401	CKR_DATA_INVALID

HSM Error	Hex Code	PKCS#11 or SFNT Defined CKR Error
LUNA_RET_SM_MULTIPLE_ACCESS_DISABLED	0x80000402	CKR_FUNCTION_NOT_SUPPORTED
LUNA_RET_SM_UNKNOWN_ACCESS_TYPE	0x80000403	CKR_ARGUMENTS_BAD
LUNA_RET_SM_BAD_ACCESS_HANDLE	0x80000404	CKR_DATA_INVALID
LUNA_RET_SM_BAD_CONTEXT_NUMBER	0x80000405	CKR_DATA_INVALID
LUNA_RET_SM_UNKNOWN_SESSION_TYPE	0x80000406	CKR_DATA_INVALID
LUNA_RET_SM_CONTEXT_ALREADY_ALLOCATED	0x80000407	CKR_DATA_INVALID
LUNA_RET_SM_CONTEXT_NOT_ALLOCATED	0x80000408	CKR_DEVICE_MEMORY
LUNA_RET_SM_CONTEXT_BUFFER_OVERFLOW	0x80000409	CKR_DEVICE_MEMORY
LUNA_RET_SM_TOSM_DOES_NOT_VALIDATE	0x8000040A	CKR_USER_NOT_LOGGED_IN
LUNA_RET_SM_ACCESS_DOES_NOT_VALIDATE	0x8000040B	CKR_USER_NOT_AUTHORIZED
LUNA_RET_MTK_ZEROIZED	0x80000531	CKR_MTK_ZEROIZED
LUNA_RET_MTK_STATE_INVALID	0x80000532	CKR_MTK_STATE_INVALID
LUNA_RET_MTK_SPLIT_INVALID	0x80000533	CKR_MTK_SPLIT_INVALID
LUNA_RET_INVALID_IP_PACKET	0x80000600	CKR_DEVICE_ERROR
LUNA_RET_INVALID_BOARD_TYPE	0x80000700	CKR_DEVICE_ERROR
LUNA_RET_ECC_NOT_SUPPORTED	0x80000601	CKR_FUNCTION_NOT_SUPPORTED
LUNA_RET_ECC_BUFFER_OVERFLOW	0x80000602	CKR_DEVICE_ERROR
LUNA_RET_ECC_POINT_INVALID	0x80000603	CKR_ECC_POINT_INVALID **
LUNA_RET_ECC_SELF_TEST_FAILURE	0x80000604	CKR_DEVICE_ERROR
LUNA_RET_ECC_UNKNOWN_CURVE	0x80000605	CKR_ECC_UNKNOWN_CURVE
LUNA_RET_HA_NOT_SUPPORTED	0x80000900	CKR_FUNCTION_NOT_SUPPORTED
LUNA_RET_HA_USER_NOT_INITIALIZED	0x80000901	CKR_OPERATION_NOT_INITIALIZED
LUNA_RET_HSM_STORAGE_FULL	0x80000902	CKR_HSM_STORAGE_FULL
LUNA_RET_CONTAINER_OBJECT_STORAGE_FULL	0x80000903	CKR_CONTAINER_OBJECT_STORAGE_FULL

HSM Error	Hex Code	PKCS#11 or SFNT Defined CKR Error
LUNA_RET_KEY_NOT_ACTIVE	0x80000904	CKR_KEY_NOT_ACTIVE
LUNA_RET_CB_NOT_SUPPORTED	0x80000a01	CKR_FUNCTION_NOT_SUPPORTED
LUNA_RET_CB_PARAM_INVALID	0x80000a02	CKR_CALLBACK_ERROR
LUNA_RET_CB_NO_MEMORY	0x80000a03	CKR_DEVICE_MEMORY
LUNA_RET_CB_TIMEOUT	0x80000a04	CKR_CALLBACK_ERROR
LUNA_RET_CB_RETRY	0x80000a05	CKR_CALLBACK_ERROR
LUNA_RET_CB_ABORTED	0x80000a06	CKR_CALLBACK_ERROR
LUNA_RET_CB_SYS_ERROR	0x80000a07	CKR_CALLBACK_ERROR
LUNA_RET_CB_HIOS_HANDLE_INVALID	0x80000a10	CKR_CALLBACK_ERROR
LUNA_RET_CB_HIOS_ID_INVALID	0x80000a11	CKR_CALLBACK_ERROR
LUNA_RET_CB_HIOS_CLOSED	0x80000a12	CKR_CALLBACK_ERROR
LUNA_RET_CB_HIOS_CANCELED	0x80000a13	CKR_CALLBACK_ERROR
LUNA_RET_CB_HIOS_IO_ERROR	0x80000a14	CKR_CALLBACK_ERROR
LUNA_RET_CB_HIOS_SEND_TIMEOUT	0x80000a15	CKR_CALLBACK_ERROR
LUNA_RET_CB_HIOS_RECV_TIMEOUT	0x80000a16	CKR_CALLBACK_ERROR
LUNA_RET_CB_HIOS_STATE_INVALID	0x80000a17	CKR_CALLBACK_ERROR
LUNA_RET_CB_HIOS_OUTPUT_BUFFER_TOO_SMALL	0x80000a18	CKR_CALLBACK_ERROR
LUNA_RET_CB_HIOS_INPUT_BUFFER_TOO_SMALL	0x80000a19	CKR_CALLBACK_ERROR
LUNA_RET_CB_HANDLE_INVALID	0x80000a20	CKR_CALLBACK_ERROR
LUNA_RET_CB_ID_INVALID	0x80000a21	CKR_CALLBACK_ERROR
LUNA_RET_CB_REMOTE_ABORT	0x80000a22	CKR_CALLBACK_ERROR
LUNA_RET_CB_REMOTE_CLOSED	0x80000a23	CKR_CALLBACK_ERROR
LUNA_RET_CB_REMOTE_ABANDONED	0x80000a24	CKR_CALLBACK_ERROR
LUNA_RET_CB_MUST_READ	0x80000a25	CKR_CALLBACK_ERROR
LUNA_RET_CB_MUST_WRITE	0x80000a26	CKR_CALLBACK_ERROR
LUNA_RET_CB_INVALID_CALL_FOR_THE_STATE	0x80000a27	CKR_CALLBACK_ERROR

HSM Error	Hex Code	PKCS#11 or SFNT Defined CKR Error
LUNA_RET_CB_SYNC_ERROR	0x80000a28	CKR_CALLBACK_ERROR
LUNA_RET_CB_PROT_DATA_INVALID	0x80000a29	CKR_CALLBACK_ERROR
LUNA_RET_LOG_FILE_NOT_OPEN	0x80000d00	CKR_LOG_FILE_NOT_OPEN
LUNA_RET_LOG_FILE_WRITE_ERROR	0x80000d01	CKR_LOG_FILE_WRITE_ERROR
LUNA_RET_LOG_BAD_FILE_NAME	0x80000d02	CKR_LOG_BAD_FILE_NAME
LUNA_RET_LOG_FULL	0x80000d03	CKR_LOG_FULL
LUNA_RET_LOG_NO_KCV	0x80000d04	CKR_LOG_NO_KCV
LUNA_RET_LOG_BAD_RECORD_HMAC	0x80000d05	CKR_LOG_BAD_RECORD_HMAC
LUNA_RET_LOG_BAD_TIME	0x80000d06	CKR_LOG_BAD_TIME
LUNA_RET_LOG_AUDIT_NOT_INITIALIZED	0x80000d07	CKR_LOG_AUDIT_NOT_INITIALIZED
LUNA_RET_LOG_RESYNC_NEEDED	0x80000d08	CKR_LOG_RESYNC_NEEDED
LUNA_RET_AUDIT_LOGIN_TIMEOUT_IN_PROGRESS	0x80000d09	CKR_AUDIT_LOGIN_TIMEOUT_IN_PROGRESS
LUNA_RET_AUDIT_LOGIN_FAILURE_THRESHOLD	0x80000d0a	CKR_AUDIT_LOGIN_FAILURE_THRESHOLD

* This error (CKR_TEMPLATE_INCONSISTENT) might be encountered when using CKDemo in a new client with firmware older than version 6.22.0. Try CKDemo option 98, sub-option 16. If it is set to "enhanced roles", try selecting it to set it to "legacy Luna roles". The setting is a toggle, and flips every time you select it.

** This error, or "unable to read public key", might be encountered when using BSAFE to encrypt data with ECC public key using curves from the Brainpool suite. As indicated on the BSAFE website (May 2012) they do not appear to support Brainpool curves. Therefore, your own applications should not attempt that combination, and you should avoid attempting to specify Brainpool curves with BSAFE ECC when using SafeNet's CKDemo utility.

Library Codes

Hex value	Decimal value	Return code/error description
0	0	OKAY, NO ERROR
0xC0000000	3221225472	PROGRAMMING ERROR: RETURN CODE
0xC0000001	3221225473	OUT OF MEMORY

Hex value	Decimal value	Return code/error description
0xC0000002	3221225474	NON-SPECIFIC ERROR
0xC0000003	3221225475	UNEXPECTED NULL POINTER
0xC0000004	3221225476	PROGRAMMING ERROR: LOGIC
0xC0000005	3221225477	OPERATION WOULD BLOCK IF ATTEMPTED
0xC0000006	3221225478	BUFFER IS TOO SMALL
0xC0000100	3221225728	OPERATION CANCEL
0xC0000101	3221225729	INVALID SLOT IDENTIFIER
0xC0000102	3221225730	INVALID DATA
0xC0000103	3221225731	INVALID PIN
0xC0000104	3221225732	NO TOKEN PRESENT
0xC0000105	3221225733	FUNCTION IS NOT SUPPORTED
0xC0000106	3221225734	NON-CRYPTOKI ELEMENT CLONE
0xC0000107	3221225735	INVALID BUFFER SIZE FOR CHALLENGE
0xC0000108	3221225736	PIN IS LOCKED
0xC0000109	3221225737	INVALID VERSION
0xC000010a	3221225738	NEEDED KEY NOT PROVIDED
0xC000010b	3221225739	USER NAME IS IN USE
0xC0000200	3221225984	INVALID DISTINGUISHED ENCODING RULES CLASS
0xC0000303	3221226243	OPERATION TIMED OUT

Hex value	Decimal value	Return code/error description
0xC0000304	3221226244	RESET FAILED
0xC0000400	3221226496	INVALID TOKEN STATE
0xC0000401	3221226497	DATA APPEARS CORRUPTED
0xC0000402	3221226498	INVALID FILENAME
0xC0000403	3221226499	FILE IS READ-ONLY
0xC0000404	3221226500	FILE ERROR
0xC0000405	3221226501	INVALID OBJECT IDENTIFIER
0xC0000406	3221226502	INVALID SOCKET ADDRESS
0xC0000407	3221226503	INVALID LISTEN SOCKET
0xC0000408	3221226504	CACHE IS NOT CURRENT
0xC0000409	3221226505	CACHE IS NOT MAPPED
0xC000040a	3221226506	OBJECT IS NOT IN LIST
0xC000040b	3221226507	INVALID INDEX
0xC000040c	3221226508	OBJECT ALREADY EXISTS
0xC000040d	3221226509	SEMAPHORE ERROR
0xC000040e	3221226510	END OF LIST ENCOUNTERED
0xC000040f	3221226511	WOULD ASSIGN SAME VALUE
0xC0000410	3221226512	INVALID GROUP NAME
0xC0000411	3221226513	NOT HSM BACKUP TOKEN

Hex value	Decimal value	Return code/error description
0xC0000412	3221226514	NOT PARTITION BACKUP TOKEN
0xC0000413	3221226515	SIM NOT SUPPORTED
0xC0000500	3221226752	SOCKET ERROR
0xC0000501	3221226753	SOCKET WRITE ERROR
0xC0000502	3221226754	SOCKET READ ERROR
0xC0000503	3221226755	CLIENT MESSAGE ERROR
0xC0000504	3221226756	SERVER DISCONNECTED
0xC0000505	3221226757	CLIENT DISCONNECTED
0xC0000506	3221226758	SOCKET WOULD BLOCK
0xC0000507	3221226759	SOCKET ADDRESS IS IN USE
0xC0000508	3221226760	SOCKET BAD FILE DESCRIPTOR
0xC0000509	3221226761	HOST RESOLUTION ERROR
0xC000050a	3221226762	INVALID HOST CERTIFICATE
0xC0000600	3221227008	NO BUFFER AVAILABLE
0xC0000601	3221227009	INVALID ENUMERATION OPTION
0xC0000700	3221227264	SSL ERROR
0xC0000701	3221227265	SSL CTX ERROR
0xC0000702	3221227266	SSL CIPHER LIST ERROR
0xC0000703	3221227267	SSL CERT VERIFICATION LOCATION ERROR

Hex value	Decimal value	Return code/error description
0xC0000704	3221227268	SSL LOAD SERVER CERT ERROR
0xC0000705	3221227269	SSL LOAD SERVER PRIVATE KEY ERROR
0xC0000706	3221227270	SSL VALIDATE SERVER PRIVATE KEY ERROR
0xC0000707	3221227271	SSL CREATE SSL ERROR
0xC0000708	3221227272	SSL LOAD CLIENT CERT ERROR
0xC0000709	3221227273	SSL GET CERTIFICATE ERROR
0xC000070a	3221227274	SSL INVALID CERT STRUCTURE
0xC000070b	3221227275	SSL LOAD CLIENT PRIVATE KEY ERROR
0xC000070c	3221227276	SSL GET PEER CERT ERROR
0xC000070d	3221227277	SSL WANT READ ERROR
0xC000070e	3221227278	SSL WANT WRITE ERROR
0xC000070f	3221227279	SSL WANT X509 LOOKUP ERROR
0xC0000710	3221227280	SSL SYSCALL ERROR
0xC0000711	3221227281	SSL FAILED HANDSHAKE
0xC0000800	3221227520	INVALID CERTIFICATE TYPE
0xC0000900	3221227776	INVALID PORT
0xC0000901	3221227777	SESSION SCRIPT EXISTS
0xC0001000	3221229568	PARTITION LOCKED
0xC0001001	3221229569	PARTITION NOT ACTIVATED

Hex value	Decimal value	Return code/error description
0xc0002000	3221233664	FAILED TO CREATE THREAD
0xc0002001	3221233665	CALLBACK ERROR
0xc0002002	3221233666	UNKNOWN CALLBACK COMMAND
0xc0002003	3221233667	SHUTTING DOWN
0xc0002004	3221233668	REMOTE SIDE DISCONNECTED
0xc0002005	3221233669	SOCKET CLOSED
0xC0002006	3221233670	INVALID COMMAND
0xC0002007	3221233671	UNKNOWN COMMAND
0xC0002008	3221233672	UNKNOWN COMMAND VERSION
0xC0002009	3221233673	FILE LOCK FAILED
0xC0002010	3221233680	FILE LOCK ERROR
0xc0002011	3221233681	FAILED TO CREATE PROCESS
0xc0002012	3221233682	USB PED NOT FOUND
0xc0002013	3221233683	USB PED NOT RESPONDING
0xc0002014	3221233684	USB PED OPERATION CANCELLED
0xc0002015	3221233685	USB PED TOO MANY CONNECTED
0xc0002016	3221233686	USB PED OUT OF SYNC
0xC0001100	3221229824	UNABLE TO CONNECT

Vendor-defined Return Codes

Code	Name
0x00000141	CKR_INSERTION_CALLBACK_NOT_SUPPORTED
0x0052	CKR_FUNCTION_PARALLEL
0x00B2	CKR_SESSION_EXCLUSIVE_EXISTS
(CKR_VENDOR_DEFINED + 0x04)	CKR_RC_ERROR
(CKR_VENDOR_DEFINED + 0x05)	CKR_CONTAINER_HANDLE_INVALID
(CKR_VENDOR_DEFINED + 0x06)	CKR_TOO_MANY_CONTAINERS
(CKR_VENDOR_DEFINED + 0x07)	CKR_USER_LOCKED_OUT
(CKR_VENDOR_DEFINED + 0x08)	CKR_CLONING_PARAMETER_ALREADY_EXISTS
(CKR_VENDOR_DEFINED + 0x09)	CKR_CLONING_PARAMETER_MISSING
(CKR_VENDOR_DEFINED + 0x0a)	CKR_CERTIFICATE_DATA_MISSING
(CKR_VENDOR_DEFINED + 0x0b)	CKR_CERTIFICATE_DATA_INVALID
(CKR_VENDOR_DEFINED + 0x0c)	CKR_ACCEL_DEVICE_ERROR
(CKR_VENDOR_DEFINED + 0x0d)	CKR_WRAPPING_ERROR
(CKR_VENDOR_DEFINED + 0x0e)	CKR_UNWRAPPING_ERROR
(CKR_VENDOR_DEFINED + 0x0f)	CKR_MAC_MISSING
(CKR_VENDOR_DEFINED + 0x10)	CKR_DAC_POLICY_PID_MISMATCH
(CKR_VENDOR_DEFINED + 0x11)	CKR_DAC_MISSING
(CKR_VENDOR_DEFINED + 0x12)	CKR_BAD_DAC
(CKR_VENDOR_DEFINED + 0x13)	CKR_SSK_MISSING
(CKR_VENDOR_DEFINED + 0x14)	CKR_BAD_MAC
(CKR_VENDOR_DEFINED + 0x15)	CKR_DAK_MISSING
(CKR_VENDOR_DEFINED + 0x16)	CKR_BAD_DAK
(CKR_VENDOR_DEFINED + 0x17)	CKR_SIM_AUTHORIZATION_FAILED

Code	Name
(CKR_VENDOR_DEFINED + 0x18)	CKR_SIM_VERSION_UNSUPPORTED
(CKR_VENDOR_DEFINED + 0x19)	CKR_SIM_CORRUPT_DATA
(CKR_VENDOR_DEFINED + 0x1a)	CKR_USER_NOT_AUTHORIZED
(CKR_VENDOR_DEFINED + 0x1b)	CKR_MAX_OBJECT_COUNT_EXCEEDED
(CKR_VENDOR_DEFINED + 0x1c)	CKR_SO_LOGIN_FAILURE_THRESHOLD
(CKR_VENDOR_DEFINED + 0x1d)	CKR_SIM_AUTHFORM_INVALID
(CKR_VENDOR_DEFINED + 0x1e)	CKR_CITS_DAK_MISSING
(CKR_VENDOR_DEFINED + 0x1f)	CKR_UNABLE_TO_CONNECT
(CKR_VENDOR_DEFINED + 0x20)	CKR_PARTITION_DISABLED
(CKR_VENDOR_DEFINED + 0x21)	CKR_CALLBACK_ERROR
(CKR_VENDOR_DEFINED + 0x22)	CKR_SECURITY_PARAMETER_MISSING
(CKR_VENDOR_DEFINED + 0x23)	CKR_SP_TIMEOUT
(CKR_VENDOR_DEFINED + 0x24)	CKR_TIMEOUT
(CKR_VENDOR_DEFINED + 0x25)	CKR_ECC_UNKNOWN_CURVE
(CKR_VENDOR_DEFINED + 0x26)	CKR_MTK_ZEROIZED
(CKR_VENDOR_DEFINED + 0x27)	CKR_MTK_STATE_INVALID
(CKR_VENDOR_DEFINED + 0x28)	CKR_INVALID_ENTRY_TYPE
(CKR_VENDOR_DEFINED + 0x29)	CKR_MTK_SPLIT_INVALID
(CKR_VENDOR_DEFINED + 0x2a)	CKR_HSM_STORAGE_FULL
(CKR_VENDOR_DEFINED + 0x2b)	CKR_DEVICE_TIMEOUT
(CKR_VENDOR_DEFINED + 0x2c)	CKR_CONTAINER_OBJECT_STORAGE_FULL
(CKR_VENDOR_DEFINED + 0x2d)	CKR_PED_CLIENT_NOT_RUNNING
(CKR_VENDOR_DEFINED + 0x2e)	CKR_PED_UNPLUGGED
(CKR_VENDOR_DEFINED + 0x2f)	CKR_ECC_POINT_INVALID
(CKR_VENDOR_DEFINED + 0x30)	CKR_OPERATION_NOT_ALLOWED
(CKR_VENDOR_DEFINED + 0x31)	CKR_LICENSE_CAPACITY_EXCEEDED
(CKR_VENDOR_DEFINED + 0x32)	CKR_LOG_FILE_NOT_OPEN

Code	Name
(CKR_VENDOR_DEFINED + 0x33)	CKR_LOG_FILE_WRITE_ERROR
(CKR_VENDOR_DEFINED + 0x34)	CKR_LOG_BAD_FILE_NAME
(CKR_VENDOR_DEFINED + 0x35)	CKR_LOG_FULL
(CKR_VENDOR_DEFINED + 0x36)	CKR_LOG_NO_KCV
(CKR_VENDOR_DEFINED + 0x37)	CKR_LOG_BAD_RECORD_HMAC
(CKR_VENDOR_DEFINED + 0x38)	CKR_LOG_BAD_TIME
(CKR_VENDOR_DEFINED + 0x39)	CKR_LOG_AUDIT_NOT_INITIALIZED
(CKR_VENDOR_DEFINED + 0x3A)	CKR_LOG_RESYNC_NEEDED
(CKR_VENDOR_DEFINED + 0x3B)	CKR_AUDIT_LOGIN_TIMEOUT_IN_PROGRESS
(CKR_VENDOR_DEFINED + 0x3C)	CKR_AUDIT_LOGIN_FAILURE_THRESHOLD
(CKR_VENDOR_DEFINED + 0x3D)	CKR_INVALID_FUF_TARGET
(CKR_VENDOR_DEFINED + 0x3E)	CKR_INVALID_FUF_HEADER
(CKR_VENDOR_DEFINED + 0x3F)	CKR_INVALID_FUF_VERSION
(CKR_VENDOR_DEFINED + 0x40)	CKR_ECC_ECC_RESULT_AT_INF
(CKR_VENDOR_DEFINED + 0x41)	CKR_AGAIN
(CKR_VENDOR_DEFINED + 0x42)	CKR_TOKEN_COPIED
(CKR_VENDOR_DEFINED + 0x43)	CKR_SLOT_NOT_EMPTY
(CKR_VENDOR_DEFINED + 0x44)	CKR_USER_ALREADY_ACTIVATED
(CKR_VENDOR_DEFINED + 0x45)	CKR_STC_NO_CONTEXT
(CKR_VENDOR_DEFINED + 0x46)	CKR_STC_CLIENT_IDENTITY_NOT_CONFIGURED
(CKR_VENDOR_DEFINED + 0x47)	CKR_STC_PARTITION_IDENTITY_NOT_CONFIGURED
(CKR_VENDOR_DEFINED + 0x48)	CKR_STC_DH_KEYGEN_ERROR
(CKR_VENDOR_DEFINED + 0x49)	CKR_STC_CIPHER_SUITE_REJECTED
(CKR_VENDOR_DEFINED + 0x4a)	CKR_STC_DH_KEY_NOT_FROM_SAME_GROUP
(CKR_VENDOR_DEFINED + 0x4b)	CKR_STC_COMPUTE_DH_KEY_ERROR
(CKR_VENDOR_DEFINED + 0x4c)	CKR_STC_FIRST_PHASE_KDF_ERROR
(CKR_VENDOR_DEFINED + 0x4d)	CKR_STC_SECOND_PHASE_KDF_ERROR

Code	Name
(CKR_VENDOR_DEFINED + 0x4e)	CKR_STC_KEY_CONFIRMATION_FAILED
(CKR_VENDOR_DEFINED + 0x4f)	CKR_STC_NO_SESSION_KEY
(CKR_VENDOR_DEFINED + 0x50)	CKR_STC_RESPONSE_BAD_MAC
(CKR_VENDOR_DEFINED + 0x51)	CKR_STC_NOT_ENABLED
(CKR_VENDOR_DEFINED + 0x52)	CKR_STC_CLIENT_HANDLE_INVALID
(CKR_VENDOR_DEFINED + 0x53)	CKR_STC_SESSION_INVALID
(CKR_VENDOR_DEFINED + 0x54)	CKR_STC_CONTAINER_INVALID
(CKR_VENDOR_DEFINED + 0x55)	CKR_STC_SEQUENCE_NUM_INVALID
(CKR_VENDOR_DEFINED + 0x56)	CKR_STC_NO_CHANNEL
(CKR_VENDOR_DEFINED + 0x57)	CKR_STC_RESPONSE_DECRYPT_ERROR
(CKR_VENDOR_DEFINED + 0x58)	CKR_STC_RESPONSE_REPLAYED
(CKR_VENDOR_DEFINED + 0x59)	CKR_STC_REKEY_CHANNEL_MISMATCH
(CKR_VENDOR_DEFINED + 0x5a)	CKR_STC_RSA_ENCRYPT_ERROR
(CKR_VENDOR_DEFINED + 0x5b)	CKR_STC_RSA_SIGN_ERROR
(CKR_VENDOR_DEFINED + 0x5c)	CKR_STC_RSA_DECRYPT_ERROR
(CKR_VENDOR_DEFINED + 0x5d)	CKR_STC_RESPONSE_UNEXPECTED_KEY
(CKR_VENDOR_DEFINED + 0x5e)	CKR_STC_UNEXPECTED_NONCE_PAYLOAD_SIZE
(CKR_VENDOR_DEFINED + 0x5f)	CKR_STC_UNEXPECTED_DH_DATA_SIZE
(CKR_VENDOR_DEFINED + 0x60)	CKR_STC_OPEN_CIPHER_MISMATCH
(CKR_VENDOR_DEFINED + 0x61)	CKR_STC_OPEN_DHNIST_PUBKEY_ERROR
(CKR_VENDOR_DEFINED + 0x62)	CKR_STC_OPEN_KEY_MATERIAL_GEN_FAIL
(CKR_VENDOR_DEFINED + 0x63)	CKR_STC_OPEN_RESP_GEN_FAIL
(CKR_VENDOR_DEFINED + 0x64)	CKR_STC_ACTIVATE_MACTAG_U_VERIFY_FAIL
(CKR_VENDOR_DEFINED + 0x65)	CKR_STC_ACTIVATE_MACTAG_V_GEN_FAIL
(CKR_VENDOR_DEFINED + 0x66)	CKR_STC_ACTIVATE_RESP_GEN_FAIL
(CKR_VENDOR_DEFINED + 0x67)	CKR_CHALLENGE_INCORRECT
(CKR_VENDOR_DEFINED + 0x68)	CKR_ACCESS_ID_INVALID

Code	Name
(CKR_VENDOR_DEFINED + 0X69)	CKR_ACCESS_ID_ALREADY_EXISTS
(CKR_VENDOR_DEFINED + 0x114)	CKR_OBJECT_READ_ONLY
(CKR_VENDOR_DEFINED + 0x136)	CKR_KEY_NOT_ACTIVE

6

High-Availability (HA) Configuration and Operation

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

- "High Availability (HA) Overview" below
- "Load Balancing" on page 202
- "Key Replication" on page 203
- "Failover" on page 204
- "Recovery" on page 207
- "Performance" on page 211
- "Standby Members" on page 213
- "Planning Your Deployment" on page 217
- "Configuring HA" on page 220
- "Using HA With Your Applications" on page 224
- "Managing and Troubleshooting Your HA Groups" on page 225
- "Adding, Removing, Replacing, or Reconnecting HA Group Members" on page 227
- "Frequently Asked Questions" on page 236

High Availability (HA) Overview

SafeNet HSM products include availability and scalability capabilities for mission-critical applications that require uninterrupted up-time. These features allow you to use the SafeNet HSM client to group multiple devices, or partitions, into a single logical group – known as an HA (High Availability) group. When an HA group is defined, cryptographic services remain available to the applications that use the client, as long as at least one member in the group remains functional and connected to the application server. In addition, the client performs load balancing among the group members, allowing many cryptographic commands to be automatically distributed across the HA group, and enabling linear performance gains for many applications.

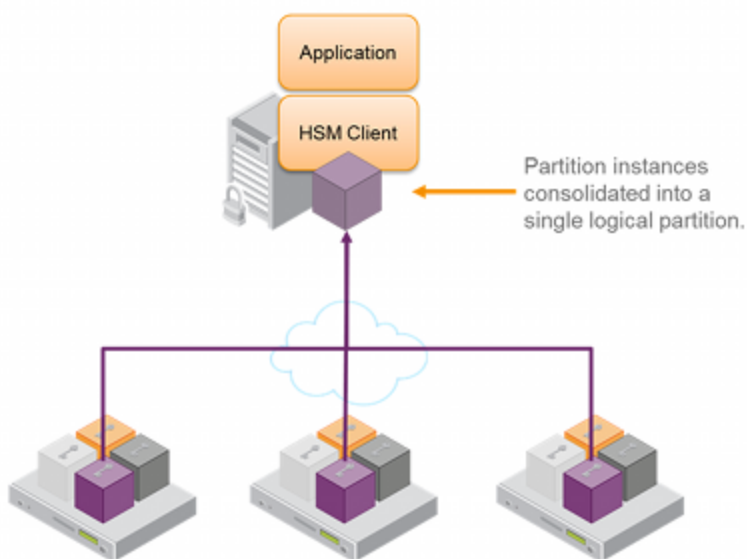
How HA is Implemented

The SafeNet high-availability (HA) and load-balancing functionality is implemented in the SafeNet HSM client, and uses the cloning¹ function to replicate/synchronize content across HA-group members. There is no direct connection between the members of an HA group. All communications between the members of an HA group are managed by the client. The HSMs and appliances are not involved and, except for being instructed to clone objects to certain HSMs during a synchronization operation, are unaware that they might be configured in an HA group. This allows you to configure HA on a per-application basis.

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

- setting automatic or manual recovery mode
- setting some HSMs as standby members
- performing various manual synchronization and recovery operations

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



Example: Database Encryption

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

¹The duplication or copying of HSM or application partition contents to other HSMs or application partitions that share the cloning domain secret. Cloning copies objects (certificates, keys, data), in a secure manner, via trusted path, from the user space on one HSM to an equivalent space on a second HSM. The trusted path can be direct connection between HSMs or application partitions on the same host, or can be via Remote Backup Protocol (RBC) between distant HSMs.

Typical Database Encryption Key Architecture

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

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

The HSM is not involved in the use of table keys. Instead it provides the strong protection of the MEK which is used to protect the table keys. Users must follow backup procedures to ensure their MEK is as durable as the database itself. Refer to the backup section of this manual for proper backup procedures.

HSM High Availability with Database Encryption

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

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

HSM Load Balancing with Database Encryption

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

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

Conclusion

The SafeNet high availability and load balancing features provide an excellent set of tools to scale applications and manage availability of cryptographic services without compromising the integrity of cryptographic keys. They do not need to be copied out of an HSM and stored in a file to achieve high levels of availability. Indeed, recovery from many failures is much more rapid with Luna's keys-in-hardware approach since each HSM maintains its own copy of all keys directly inside it. A broad range of deployment options are supported that allow solution architects to achieve the

availability needed in a manner that optimizes the cost and performance without compromising the assurance of the solution.

Load Balancing

The default behavior of the client library is to attempt to load-balance the application's cryptographic requests across each active member of an HA group. Any standby members in the HA group are not used to perform cryptographic operations, and are therefore not part of the load-balancing scheme (see ["Standby Members" on page 213](#)).

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

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

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

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

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

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

Example

There is no "master" HSM appliance in the SafeNet Network HSM HA model. Where you might see or hear mention of a "Primary" member, that refers only to the member that happens to be the first on the configuration list. If you edit the list to place the name of a different SafeNet Network HSM on top, then that becomes the new HA Group "primary" member.

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

In practice, that means the primary member gets all the requests until the volume reaches a level that saturates the ability of the primary, or a blocking request from another source prevents acceptance of new requests. Therefore, on a 7000 signings/second SafeNet Network HSM doing exclusively 1024-bit RSA signings, your client would need to have approximately 30 simultaneous threads offering a total of nearly 7000 requests per second before the second member would begin seeing any requests. In other words, until the primary is fully occupied, the HA group looks like it is operating as a "hot-standby" arrangement.

The numbers above are ideal, of course. If you add network latency, or if you increase the key-size, or if you interleave other crypto operations, then the numbers must drop for the individual member, and the secondary member becomes part of the overall performance. And a third member, if you have a third active member in your group, and so on.

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

Key Replication

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

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

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

The cloning protocol is such that it must be invoked separately for each object to be cloned and the sequence of calls required to implement the protocol must be issued by an authorized client library (residing on a client platform that has

been authenticated to each of the SafeNet HSMs involved in the HA group). This ensures that the use of the cloning function calls is controlled and the protocol cannot be misused to permit the unauthorized transfer of objects to or from one of the partitions in the HA group.

Manual Synchronization

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

Failover

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

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

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

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

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

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

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

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

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

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

Multi part operations do NOT fail over. The entire operation returns a failure. 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.

Reaction to Failures

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

- Failure at the HSM or appliance
 - HSM card failure
 - HSM re-initialization
 - Deactivated partition
 - Power failure of a member
 - Reboot of member
 - NTL failure
 - STC failure
- Failure at the client
 - Power failure of the client
 - Reboot of client
 - Network keepalive failure
- Failure between client and group members
 - Network failure near the member appliance
(so only one member might disappear from client's view)
 - Network failure near the client
(client loses contact with all members)

HSM-Side Failures

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

Temporary

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

Permanent

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

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

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

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

Client-Side Failures

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

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

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

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

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

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

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

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

Recovery

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

Automatic recovery

With automatic recovery, the client automatically performs periodic recovery attempts while a member is failed. The frequency of these checks is adjustable and the number of re-tries can be limited. Each time a reconnection is attempted, one application command experiences a slight delay while the client attempts to recover. As such, the retry frequency cannot be set any faster than once per minute. Even if a manual recovery process is selected, the application does not need to be restarted. Simply run the client recovery command and the recovery logic inside the client makes a recovery attempt the next time the application uses the HSM. As part of recovery, any key material created while the member was offline is automatically replicated to the recovered unit.

Failed units

Sometimes a failure of a device is permanent. In this event, the only solution is to deploy a new member to the group. In this case, you can remove the failed unit from the HA group, add a new device to the group and then start the recovery process. The running clients automatically resynchronize keys to the new member and start scheduling operations to it. See ["Adding, Removing, Replacing, or Reconnecting HA Group Members"](#) on page 227 for more information.

Manual recovery

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

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

Usage

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

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

Recovery Conditions

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

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

HA recovery takes place if the following conditions are true:

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

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

The HA recovery logic makes its first attempt at recovering a failed member when your application makes a call to its HSM (the group). That is, an idle client does not start the recovery-attempt process.

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

Enabling and Configuring Autorecovery

In previous releases, Autorecovery was not on by default, and needed to be explicitly enabled with `vtl haAdmin - autorecovery` command.

Beginning with SafeNet HSM release 6.0, HA autorecovery is automatically enabled when you set the recovery retry count using the LunaCM command "[hagroup retry](#)" on page 1 in the *LunaCM Reference Guide*. Use the command "[hagroup interval](#)" on page 1 to specify the interval, in seconds, between each retry attempt. The default is 60 seconds.

Failure of All Members

If all members of an HA group were to fail, then all logged-in sessions are gone, and operations that were active when the last group member went down, are terminated. It is not currently possible from the SafeNet Network HSM perspective to resume the client's state unassisted when the members are restarted. However, if the client application is able to recover all that state information, then it is not necessary to restart or re-initialize in order to resume client operations with the SafeNet Network HSM HA group.

Automatic Reintroduction

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

Synchronization

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

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

Effect of PED Operations

PED operations block cryptographic operations, so that while a member of an HA is performing a PED operation, it will appear to the HA group as a failed member. When the PED operation is complete, failover and recovery HA logic are invoked to return the member to normal operation.

Effect of Application Restarts

If an HA group member fails and an application restarts before the failed member recovers, it is not possible to recover that device until you restart the application again.

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. 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 while all X members are available.

Network failures

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

- While the client application is active, and one HA group member is connected and active, other members can automatically resume in the HA group as long as retries have not stopped.
- If all members fail or if the client does not have a network connection to at least one group member, then the client application must be restarted.

Process interaction

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

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

- In a period of no activity by client or appliance, the appliance's TCP will wonder if the client is still there, and will send a packet after 3 minutes of silence.
- If that packet is acknowledged, the 3 minute TCP timer restarts, and the cycle repeats indefinitely.
- If the packet is NOT acknowledged, then TCP sends another after approximately 45 seconds, and then another after a further 45 seconds. At the two minute mark, with no response, the connection is considered dead, and higher levels are alerted to perform their cleanup.

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

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

While the HSM processes the request, appliance-side NTLS sends a "keep-alive PING" every two seconds, until the HSM returns the answer, which NTLS then conveys across the link to the requesting client. NTLS (nor any layer above) does not perform any interpretation of the ping.

It simply drops a slow, steady trickle of bytes into the pipe, to keep the TCP layer active. This normally has little effect, but if your client requests a lengthy operation like (say) an 8192-bit keygen, then the random-number-generation portion of that operation could take many minutes to complete, during which the HSM would legitimately be sending nothing back to the client. The NTLS ping ensures that the connection remains alive during long pauses.

Configuration settings

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

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

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

A minute later, the HA client tries to contact the member that failed to reply.

If the connection is successfully re-established, the errant appliance resumes working in the group, being assigned application calls as needed (governed by application workload and HA logic).

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

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

Active versus Passive Autorecovery on a SafeNet Network HSM

Prior to SafeNet HSM release 6.2, recovery was a passive process, and was triggered only in conjunction with a PKCS#11 library call. This permitted a rare, but possible condition where all members of an HA group experienced temporary failure within a short time of each other, and were unable to synchronize and perform failover. If a "secondary" member had dropped out of the HA group, and come back, the library was unaware that it needed synchronization until the active member experienced some kind of failure and it was discovered that the next member was not up-to-date, at which point it was too late for synchronization. This case is addressed by making the recovery task an active background task, independent of library calls to a particular member.

From SafeNet HSM 6.2 and later, the PKCS#11 stack is no longer tasked with the autorecovery function, and instead a new HA Active Recovery Thread (ARCT) is introduced. The ARCT sends a non-session-based message that is processed by NTLS. This allows recovery as soon as a failed member returns, and does not wait for a PKCS#11 operation. Thus, if a failed member returns to duty before an active member fails, then synchronization occurs immediately, and the secondary member is ready to take over from the active member if that now fails.

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

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

Active recovery is optional

Active recovery mode is optional, and is controlled by the LunaCM **hagroup recoverymode** command.

Solaris (and other Unix)

Due to a problem in the TCP/IP configuration of some Solaris systems, inconvenient delays may have been experienced with some Solaris clients.

The problem occurred if an application was started on a Solaris client while one or more expected SafeNet Network HSM appliances is unavailable. The Solaris client machine experienced a considerable delay (minutes) before the remaining SafeNet Enterprise HSMs could be seen and used by the application. This was a TCP/IP setup issue in Solaris, in which the system attempted to set up sockets for each expected connection, and retried the unsuccessful attempts until timeout, before permitting successful connections to proceed.

To control this problem, the client-side library now imposes a ten-second retry window per expected appliance, and then moves on. (Thus, if your Client was configured to use three SafeNet Network HSM appliances, and two of them were unavailable, the Client would retry the first missing appliance for ten seconds, then the second missing appliance for a further ten seconds, for a total of twenty seconds of retries, before resuming operation with the remaining available appliance). This applies to Linux and Unix variants.

For Windows, the per-appliance timeout is 24 seconds.

Performance

For repetitive operations, like a high volume of signings using the same key, an HA group can expand SafeNet Network HSM performance in linear fashion as HA group members are added. HA groups of 16 members have undergone long-term, full-throttle testing, with excellent results.

Do keep in mind that simply adding more and more SafeNet Network HSM appliances to an HA group is not an infallible recipe for endless performance improvement. For best overall performance, all HA group members should be driven near their individual performance "sweet spot", which for SafeNet Network HSM 5.2 and later is around 30 simultaneous threads per HSM. If you assemble an HA group that is considerably larger than your server(s) can drive, then you might not achieve full performance from all.

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

Maximizing Performance

SafeNet Network HSM 6.x in HA can provide performance improvement for asymmetric single-part operations. Gigabit ethernet connections are recommended to maximize performance. For example, we have seen as much as a doubling of asymmetric single-part operations in a two-member group in a controlled laboratory environment (without crossing subnet boundaries, without competing traffic or other latency-inducing factors).

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

Single-part cryptographic operations are load-balanced by the SafeNet HA functionality under most circumstances (see note on PE1746¹Enabled setting). Load-balancing these operations provides both scalability (better net throughput of operations) and redundancy by supporting transparent fail-over.

Performance is Dependent on the Type of Operation

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

With HA

- One encryption (to wrap the key)
- One decryption in the HSM (to unwrap the key)
- Object Creation on the HSM (the unwrapped key is created and stored as a key object)
- Key Replication happens for HA
 - RSA 4096-bit operation used to derive a shared secret between HSM
 - Encryption of the key on the primary HA member using the shared secret
 - Decryption of the key on the secondary HA member hsm using the shared secret
 - Object Creation on the second HA member
- One encryption (uses the unwrapped key object to encrypt the data)

Without HA

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

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

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

HA and the PE1746Enabled Setting

The SafeNet HSM client accepts a configuration file entry known as “PE1746Enabled”. This configures the way SafeNet HSM handles symmetric encryption and decryption operations for certain algorithms – namely ECB and CBC modes of AES and TDES. By default (beginning with release 5.4) an entry is always present in the [Misc] section of the configuration file, and its value is set to “PE1746Enabled=0”, or unset.

To set this configuration option, “PE1746Enabled=1”.

¹crypto integrated circuit within the K6 HSM (the stand-alone Luna PCI-E, and the HSM inside the Luna SA appliance).

When set, this value configures the library to use fast-path cryptography directly to symmetric encryption engines. This has the advantage of enabling high performance bulk crypto performance, but has the disadvantage of creating a direct context between the client library and the engine. This means that the library cannot easily load-balance operations across HSMs. This mode should be used only by applications that perform large data encryption operations (>1K data sizes).

When `PE1746Enabled=0`, the library uses its standard command path to the HSM. The advantage of this is that all single-part cryptographic operations can be load-balanced. The disadvantage is lower performance for larger data sizes. Applications should maintain this setting whenever possible to ensure the scalability and fail-over advantages.

In summary:

- when `PE1746Enabled=1` load-balancing is not used for symmetric cryptographic operations; instead all symmetric operations are directed at the client's primary member -- you see better performance, but no scalability across HSMs.
- when `PE1746Enabled=0` all single-part cryptographic operations (with data size less-than-or-equal-to 1K) are load-balanced.

A single-part crypto operation is typically one that has small data sizes (< 1Kb), but is also dependent on how the library makes its API calls (PKCS #11 supports explicit multi-part API calls through the use of `C_EncryptUpdate` and `C_DecryptUpdate`). When an application uses the "Update" APIs the cryptographic operation is, by definition, multi-part. When the application does not use these APIs (i.e. uses `C_EncryptInit` followed by `C_Encrypt`) then an operation is single-part up to a 64KB data size.

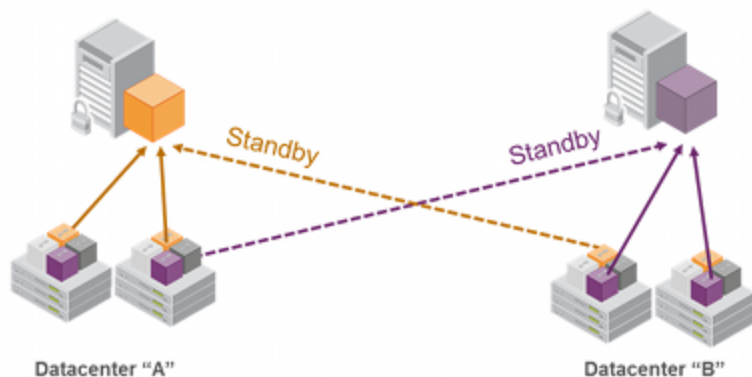
Additionally, the HSM has a limit of 1000 contexts for SafeXcel 1746 operations, which is a consideration when many client threads are involved, and depends upon the number of concurrent threads.

Whenever possible, run your application with `PE1746Enabled=0`.

Standby Members

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

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



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



Note: STANDBY BEHAVIOR - Standby members become active only to keep the group alive. In an HA group that includes more than one standby member, if all active members go down/off-line, only one standby member becomes active in the group. Other standby members remain on standby until/unless that first standby member goes down, at which point the next standby member becomes active.

In other words, in an HA group, the load-sharing and redundancy capability is as large as all the active members, but if all active members become unavailable to the application, then the group load-sharing and redundancy consists of one member, even if other standby members are still available.

To set an HSM to standby status

In "Configuring HA" on page 220, we created an HA group with label "myHAGroup" and group number 742276409, with two active members, serial number 65003001 and serial number 65005001.

1. Create a third member, as previously described, and add it to the HA group.

```
lunacm: hagroup addmember -group 742276409 -serialnumber 66010002
```

Member 65005002 successfully added to group 742276409.

Group configuration is:

```
HA Group Label: myHAGroup
HA Group Number: 742276409
HA Group Slot ID: Not Available
Synchronization: enabled
Group Members: 65003001, 65005001, 66010002
Needs sync: no
```

Slot #	Member S/N	Member Label	Status
--------	------------	--------------	--------

```

=====
0      65003001      sa175legpar1      alive
1      65005001      sa172legpar1      alive
1      66010002      sa172legpar1      alive
=====

```

Command Result : No Error

LunaCM v6.0.0 - Copyright (c) 2006-2015 SafeNet, Inc.

Available HSMs:

```

Slot Id -> 0
Label -> sa175legpar1
Serial Number -> 65003001
Model -> LunaSA
Firmware Version -> 6.22.0
Configuration -> Luna User Partition, No SO (PW) Signing With Cloning Mode
Slot Description -> Net Token Slot

```

```

Slot Id -> 1
Label -> sa172legpar1
Serial Number -> 65005001
Model -> LunaSA
Firmware Version -> 6.22.0
Configuration -> Luna User Partition, No SO (PW) Signing With Cloning Mode
Slot Description -> Net Token Slot

```

```

Slot Id -> 2
Label -> sa177legpar1
Serial Number -> 66010002
Model -> LunaSA
Firmware Version -> 6.22.0
Configuration -> Luna User Partition, No SO (PW) Signing With Cloning Mode
Slot Description -> Net Token Slot

```

```

Slot Id -> 3
HSM Label -> myHAGroup
HSM Serial Number -> 742276409
HSM Model -> LunaVirtual
HSM Firmware Version -> 6.22.0
HSM Configuration -> Luna Virtual HSM (PW) Signing With Cloning Mode
HSM Status -> N/A - HA Group

```

Current Slot Id: 0

2. Set the member to standby status.

```

lunacm: hagroup addstandby -group 742276409 -serialnumber 66010002
Member 65005002 successfully added to group 742276409.

```

Group configuration is:

```

HA Group Label: myHAGroup
HA Group Number: 742276409
HA Group Slot ID: Not Available
Synchronization: enabled
Group Members: 65003001, 65005001, 66010002

```

Needs sync: no

Slot #	Member S/N	Member Label	Status
=====	=====	=====	=====
0	65003001	sa175legpar1	alive
1	65005001	sa172legpar1	alive
1	66010002	sa172legpar1	standby

Command Result : No Error

LunaCM v6.0.0 - Copyright (c) 2006-2015 SafeNet, Inc.

Available HSMs:

```
Slot Id -> 0
Label -> sa175legpar1
Serial Number -> 65003001
Model -> LunaSA
Firmware Version -> 6.22.0
Configuration -> Luna User Partition, No SO (PW) Signing With Cloning Mode
Slot Description -> Net Token Slot
```

```
Slot Id -> 1
Label -> sa172legpar1
Serial Number -> 65005001
Model -> LunaSA
Firmware Version -> 6.22.0
Configuration -> Luna User Partition, No SO (PW) Signing With Cloning Mode
Slot Description -> Net Token Slot
```

```
Slot Id -> 2
Label -> sa177legpar1
Serial Number -> 66010002
Model -> LunaSA
Firmware Version -> 6.22.0
Configuration -> Luna User Partition, No SO (PW) Signing With Cloning Mode
Slot Description -> Net Token Slot
```

```
Slot Id -> 3
HSM Label -> myHAGroup
HSM Serial Number -> 742276409
HSM Model -> LunaVirtual
HSM Firmware Version -> 6.22.0
HSM Configuration -> Luna Virtual HSM (PW) Signing With Cloning Mode
HSM Status -> N/A - HA Group
```

Current Slot Id: 0

Planning Your Deployment

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

HA Group Members

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

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

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

Mix and Match Software Is Not Supported

All SafeNet Network HSM appliances in an HA group must be at the same revision level. If you have SafeNet Network HSM units at different version levels, perform updates as necessary, before attempting to create an HA group -this applies to the system software version, not to the HSM firmware, which **can** differ among group members.

Mix and Match Firmware Is Not Recommended

Generally, keep all HA members at the same firmware version. As well, all members should have the same optional capability updates applied. If mismatches are permitted among members, synchronization might be disrupted if your application attempts to use a mechanism or a capability that not all members support. In the previous section, we indicate that HSM firmware can differ between members of an HA group, but this is not intended for ongoing operation; rather, it allows you to keep all members within a group while you individually update their firmware, to ensure minimal disruption during the updates.

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 HSMs 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.

HA Group Members Must Not Be on the Same Appliance

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

Running HA on a group of SIM SafeNet Network HSM appliances

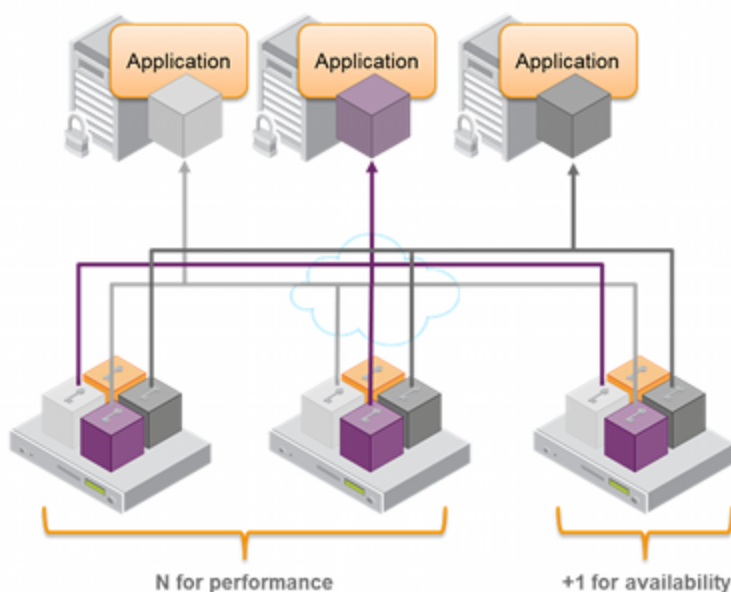
SIM replication is supported. HA will work, but key replication must be performed manually, that is, key creation in such an environment will fail to replicate.

Running HA on a group of export SafeNet Network HSM appliances

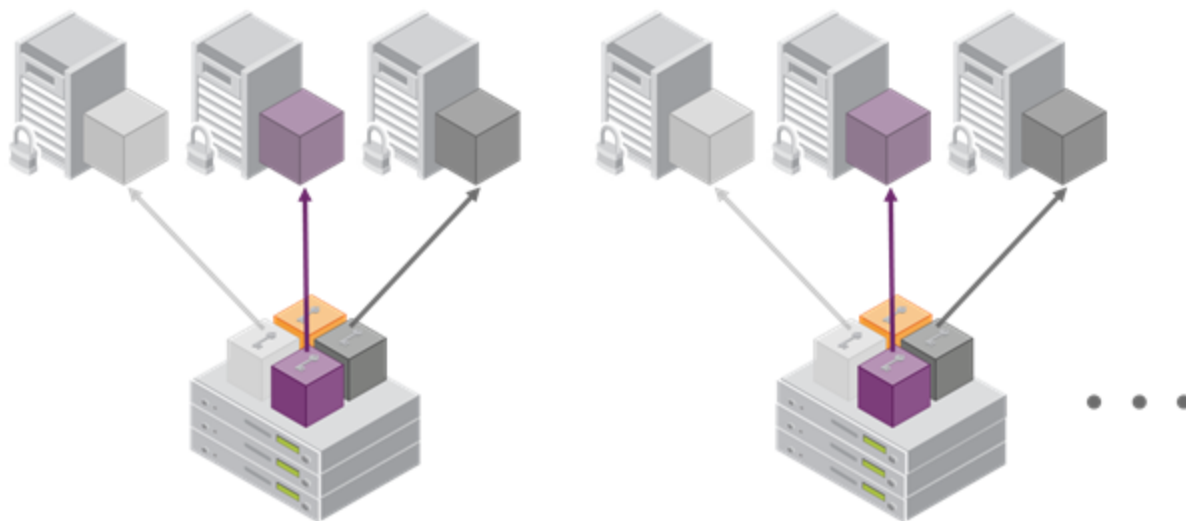
This configuration is supported, although you cannot clone/replicate private keys.

High Availability Group Sizing

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



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



Network Requirements

The network topography of the HA group is generally not important to the proper functioning of the group. As long as the client has a network path to each member the HA logic will function. Keep in mind that having a varying range of latencies between the client and each HA member causes a command scheduling bias towards the low-latency members. It also implies that commands scheduled on the long-latency devices have a larger overall latency associated with each command. In this case, the command latency is a characteristic of the network; to achieve uniform load distribution ensure that latencies to each device in the group are similar (or use standby mode). Gigabit Ethernet network connections are recommended.

Upgrading and Redundancy and Rotation

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

If the application is so critical that you cannot permit that much scheduled downtime, then you can set up a second complete set of Client computer and associated HA group. One set can service the application load while the other set

is being upgraded or otherwise maintained. For such up-time-critical applications, you might already have such a backup set of Client-plus-HA-group that you would rotate in and out of service during regular maintenance windows.

Configuring HA

For this section you need at least two SafeNet Network HSM appliances with PED Authentication, or two with Password Authentication. You may not use Password Authenticated SafeNet Network HSM and PED Authenticated SafeNet Network HSM simultaneously in an HA group.

Set up Appliances for HA

Follow these steps to set up an HA group:

1. Perform the network setup on your two HA units (for a description of the standard procedure, see ["Configuring the SafeNet Appliance Network Settings" on page 1](#) in the *Configuration Guide*). For this example, the appliances are designated sa1751 and sa172 and their HSMs have the same names, respectively.
2. Ensure that the **Allow Cloning** and **Allow Network Replication** policies are "On" in **hsm showPolicies** (and if not, then set them with **hsm setPolicy**). If your HSMs do not have the cloning option, then they will use the SIM or Key Export functionality to backup to (and restore from) a file, rather than a hardware Backup token).
3. Initialize the HSMs on your SafeNet Network HSM appliances (See ["About Initializing a Password-Authenticated HSM"](#) or See ["Initializing a PED-Authenticated HSM"](#) in the *Configuration Guide*). They must have the same cloning domain – that is, they must share the same red, domain PED Key if they are PED-authenticated , or they must share the same domain string if they are password-authenticated.
4. Create a partition on each SafeNet Network HSM. They need not have the same labels, but must have the same password. For this example, the Partitions are sa175legpar1 (on sa175) and sa172legpar1(on sa172).
5. Use the **partition changePw** command to change the Partitions' passwords so that they match.
By making the client partition challenge password the same on both partitions (on both SafeNet Network HSM appliances), you allow your clients to use that one secret when addressing the virtual partition (which includes both real partitions).
6. Make a note of the serial number of each Partition created on each SafeNet Network HSM (use **partition show**). For this example:
 - sa175 - sa175legpar1 - serial number 65003001 - password userpin
 - sa172 - sa172legpar1 - serial number 65005001 - password userpin.
7. [OPTION] Ensure that each Partition is Activated and AutoActivated (see ["About Activation and Auto-Activation" on page 268](#) - applies to SafeNet Network HSM with PED Authentication), so that it can retain/resume its "Activate" (persistent login) state through any brief power failure or other interruption.

Register Clients with SafeNet Network HSM HA

Proceed with normal client setup (see ["Creating a Network Link Between the Client and the Partition" on page 1](#) in the *Configuration Guide*). Register your client computer with both SafeNet Enterprise HSMs (this example is using just two HSM appliances; obviously, you would configure and register however many HSM appliances you wish to use in your own situation).

- On sa175, assign sa175legpar1 to ClientX (you would replace "ClientX" with the actual name of your Client computer).

- On sa172, assign sa172legpar1 to ClientX, as well (repeat if you have more SafeNet Enterprise HSMs and Partitions to include in the HA group).

At this point, you have completed a normal single-client, multiple HSM appliance setup.

Now proceed to create the HA group.

Create the HA Group



Note: Your LunaCM instance needs to update the **Chrystoki.conf** (Linux/UNIX) or **crystoki.ini** file (Windows) when setting up or reconfiguring HA. Ensure that you have sufficient privileges.

After creating partitions on (at least) two SafeNet appliances, and setting up NTLS between those partitions and your client, use LunaCM to configure HA on your client.

- Use the **hagroup addmember** command to create a new HA group on the client, which requires:

- a Label for the group (do NOT call the group just "HA").
- the Serial number of the first partition OR the slot number of the first partition.
- the password for the partition.

Lunacm also generates and assigns a Serial Number to the group itself:

```
lunacm:> hagroup addMember -group myHAGroup -serialNumber 65003001 -label -password userpin
New group with label "myHAGroup" created with group number 742276409.
```

Group configuration is:

```
HA Group Label:  myHAGroup
HA Group Number: 742276409
HA Group Slot ID: Not Available
Synchronization: enabled
Group Members:   65003001
Needs sync:      no
```

Slot #	Member S/N	Member Label	Status
=====	=====	=====	=====
0	65003001	sa175legpar1	alive

Command Result : No Error

LunaCM v6.0.0 - Copyright (c) 2006-2015 SafeNet, Inc.

Available HSMs:

```
Slot Id -> 0
Label -> sa175legpar1
Serial Number -> 65003001
Model -> LunaSA
Firmware Version -> 6.22.0
Configuration -> Luna User Partition, No SO (PW) Signing With Cloning Mode
Slot Description -> Net Token Slot

Slot Id -> 1
Label -> sa172legpar1
```

```

Serial Number ->      65005001
Model ->              LunaSA
Firmware Version ->   6.22.0
Configuration ->      Luna User Partition, No SO (PW) Signing With Cloning Mode
Slot Description ->    Net Token Slot

Slot Id ->            3
HSM Label ->          myHAGroup
HSM Serial Number ->   742276409
HSM Model ->           LunaVirtual
HSM Firmware Version -> 6.22.0
HSM Configuration ->   Luna Virtual HSM (PW) Signing With Cloning Mode
HSM Status ->          N/A - HA Group

```

Current Slot Id: 0



Note: The above is for Password-authenticated SafeNet HSMs. For PED-authenticated HSMs, have a SafeNet PED connected, the partition already activated, and provide the partition challenge secret as the password (must be the same for all members).

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

```

VirtualToken = {
VirtualToken00Members = 65003001;
VirtualToken00SN = 742276409;
VirtualToken00Label = myHAGroup;
}

```



CAUTION: Never insert TAB characters into the `chrystoki.ini` (Windows) or `crystoki.conf` (UNIX) file.

3. Use the **hagroup addmember** command to add another member to the HA group, that member being Partition2 on Luna2:

```
lunacm:> hagroup addMember -group myHAGroup -serialNumber 65005001 -password userpin
```

Member 65005001 successfully added to group 742276409.

New group configuration is:

HA Group Number: 742276409

HA Group Label: myHAGroup

Group Members: 65003001, 65005001

Needs sync: no

Group configuration is:

HA Group Label: myHAGroup

HA Group Number: 742276409

HA Group Slot ID: Not Available

Synchronization: enabled

Group Members: 65003001, 65005001

Needs sync: no

Slot #	Member S/N	Member Label	Status
=====	=====	=====	=====
0	65003001	sa175legpar1	alive
1	65005001	sa172legpar1	alive

Command Result : No Error

LunaCM v6.0.0 - Copyright (c) 2006-2015 SafeNet, Inc.

Available HSMs:

```
Slot Id -> 0
Label -> sa175legpar1
Serial Number -> 65003001
Model -> LunaSA
Firmware Version -> 6.22.0
Configuration -> Luna User Partition, No SO (PW) Signing With Cloning Mode
Slot Description -> Net Token Slot
```

```
Slot Id -> 1
Label -> sa172legpar1
Serial Number -> 65005001
Model -> LunaSA
Firmware Version -> 6.22.0
Configuration -> Luna User Partition, No SO (PW) Signing With Cloning Mode
Slot Description -> Net Token Slot
```

```
Slot Id -> 3
HSM Label -> myHAGroup
HSM Serial Number -> 742276409
HSM Model -> LunaVirtual
HSM Firmware Version -> 6.22.0
HSM Configuration -> Luna Virtual HSM (PW) Signing With Cloning Mode
HSM Status -> N/A - HA Group
```

Current Slot Id: 0

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

```
VirtualToken = {
VirtualToken00Members = 65003001, 65005001;
VirtualToken00SN = 742276409;
VirtualToken00Label = myHAGroup;
}
```

5. Use the command **hagroup synchronize -group <grouplabel> -password <password> -enable** when you are ready to replicate data between/among all members of the HA group.

If you have additional members to add, you might wish to wait until you have added them before synchronizing to save time by avoiding multiple synchronizations. The 'synchronize' command replicates all objects on all partitions across all other partitions. As there are no objects on our newly created partitions yet, we do not need to run this command.



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

Verification Steps

6. We have the two physical slots on SafeNet HSM sa175 and SafeNet HSM sa172, and now a third virtual slot which points at both physical slots at once, via load balancing. To test your HA setup, run multitoken against slot 3:
`./multitoken -mode rsasigver -key 1024 -slots 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3`



Note: (Each of the “3”s in the above sample invokes one thread performing the selected signing operation.)

If you are satisfied that your HA setup is working, then you can begin using your application against the HA “slot” label (which, in the example above, was “myHAGroup”). If you have included more SafeNet HSM application Partitions in your HA group, then the virtual slot assignment will differ accordingly, but that doesn’t matter to your application, because the application should be invoking the label, not a particular slot-number.

HA Standby Mode [optional]

If you wish to add an additional member that will be designated a standby member, and not a regular participant in the group, see “Standby Members” on page 213.

Using HA With Your Applications

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

HAOnly

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

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

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

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

Also, from the perspective of the Client, a member of the HA group can fail and, with “haonly” set, the slot count does not change. If “haonly” is not set, and both virtual and physical slots are visible, then failure of unit number 1 causes unit

number 2 to become slot 1, and so on. That could cause problems if your application is not designed to deal gracefully with such a change.

Key Generation

An application that continuously generates key material will need to have its HA group carefully selected. Contact SafeNet support for help in architecting your HA group for these applications.

Example

Multi-token is a general-purpose demonstration/exercise tool for SafeNet HSMs. It is not optimized for all tasks. If you run the extract/insert options (for SIM) in multitoken against SafeNet Network HSM with several threads against the HA slot, performance appears to be about 10 times slower than in non-HA single slot mode.

The reason is that in this mode multitoken continuously creates session objects that need to be replicated to the additional physical HA slots. This creates overhead that does not exist in single slot mode. For optimum real-life performance, your applications should avoid this approach.

Application Object Handles

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

C_FindObjects Behavior

If your applications use the C_FindObjects function to list the objects on an HA slot, the client will iterate between each member of the HA group to determine the list of objects. That is, C_FindObjects queries each HSM/partition in the group, for each object, and creates a virtual object list that contains a virtual object handle for each object found. Because the client must iterate through each HA member, performance is much slower than is observed when querying a single HSM or partition, and degrades linearly for each additional HA group member. For small HA groups containing only a few objects, this is not an issue. If, however, your HA group contains a large number of objects, or consists of several members, application performance may suffer.

To mitigate performance degradation when using the C_FindObjects function to list the objects on an HA slot, we recommend that you structure your applications such that they use C_FindObjects to iterate over the members of the HA group only once, and cache the resulting object list. Subsequent C_FindObjects function calls would then use the cached object list to look up existing objects, and only iterate over the members of the HA group to find new objects, which are appended to the cached object list.



Note: The cached object list is ephemeral, and only exists for the current session. If you restart the application, you must recreate the object list cache. Best practice is to execute C_FindObjects to create the cached object list at application start up.

Managing and Troubleshooting Your HA Groups

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

Slot Enumeration

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

```
bash-3.2# ./vtl listslot
```

Number of slots: 11

The following slots were found:

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



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

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

Determining Which Device is in Use

Use the "ntls show" command.

Determining Which Devices are Active

CA extension call "CA_GetHState" lists all active devices. The LunaCM **hagroup listgroup** command also lists members.

Duplicate Objects

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

There are TWO implications from this situation:

- One implication is that repeated duplication (perhaps an application that performs periodic backups, and restores using the 'add' option rather than 'replace') could cause the Partition to reach the maximum number of Partition

objects while seemingly having fewer objects. If the system ever tells you that your Partition is full, but HA says otherwise, then use a tool like ckdemo that can view the "physical" slots directly (as opposed to the HA slot) on the HSM, and delete any objects that are unnecessary.

- A second implication is that the HA feature uses object fingerprints to match different instances of an object on different physical HSMs. This can result in error messages if your application does not properly create and destroy session objects, and perhaps creates an object identical to one which has been removed in a separate concurrent session. The problem is self-correcting, but the flurry of error messages could be worrying if you don't understand where they are coming from.

Adding, Removing, Replacing, or Reconnecting HA Group Members

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

Adding or Removing an HA Group Member

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

- `hagroup addmember`
- `hagroup addstandby`
- `hagroup removemember`
- `hagroup removestandby`

See "[hagroup](#)" on [page 1](#) in the *LunaCM Command Reference Guide* for detailed descriptions and syntax for each `hagroup` command.



Note: You must restart the application to have the added or removed member recognized.

Reconnecting an Off-line Unit

In HA mode, if an HSM appliance goes off-line/drops-out (due to failure, maintenance, or other reason), the application load is spread over the remaining HSM Partitions on appliances in the HA Group. When the unit is restarted, the application does **not** need 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), if it was powered off for more than a short outage, you must re-activate the Partitions before they can be re-included into the HA Group.

The following two reconnection scenarios are available:

To recover the same group member

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

```
lunacm:> ha -recover -group <group_name>
```

To replace a failed group member with a new appliance

1. Configure the new SafeNet Network HSM as follows:
 - name it differently from the failed member appliance, The name must be different to avoid any possibility of

conflict between the old and new SSL certificates, which incorporate the hostnames of the respective appliances.

- make it part of the same cloning domain as others in the HA group. At initialization, the HSM gets 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 (password, autoActivation, auto MofN, client assignments, etc.).
3. Do not delete the failed SafeNet Network HSM member from the Chrystoki.conf (Unix/Linux) or Crystoki.ini (Windows) configuration file.
4. Determine the serial number of the failed member partition.
5. Retrieve the server certificate of the new SafeNet Network HSM.
6. Replace the failed SafeNet Network HSM with the new one using the following VTL command:
`vtl replaceServer -o <oldServerName> -n <newServerName> -c <newServerCertFile>`
7. Add the new partition of the new SafeNet Network HSM to the HA group using the relevant command below:
 - `lunacm:> ha -addMember -group <group number> -serialNum <serialnumber> -password <password>`
 - `lunacm:> ha -addMember -group <group number> -slot <slotnumber> -password <password>`
8. Remove the failed member from the HA group, using the relevant command below:
 - `lunacm:> ha -removeMember -group <groupNumber> -serialNum <serialnumber>`
 - `lunacm:> ha -removeMember -group <groupNumber> -slot <slotnumber>`
9. Do not perform a manual re-synchronization between the members. Instead, use the following command:
 - `lunacm:> ha -recover -group <group_name>`

Replacing a Failed SafeNet Network HSM

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

Chrystoki.ini before client-side certificate creation

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

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

[Luna]
DefaultTimeOut=500000
```



```

PEDTimeout1=100000
PEDTimeout2=200000
PEDTimeout3=10000

```

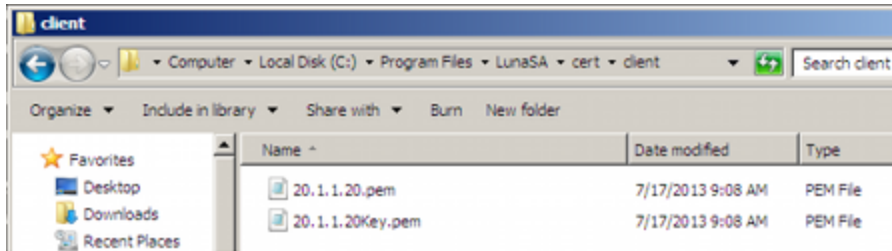
```

[CardReader]
RemoteCommand=1

```

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

Generated client certificates



Chrystoki.ini after client-side certificate creation

```

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

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

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

[CardReader]
RemoteCommand=1

```

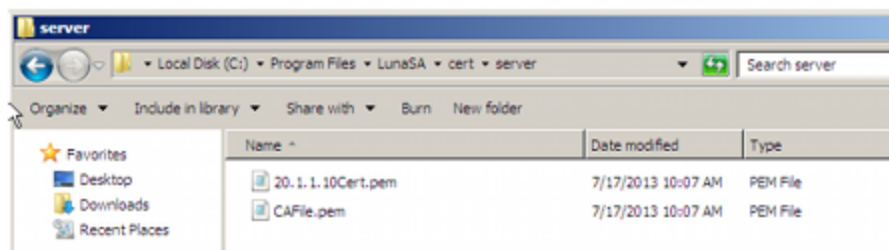
2. Copy SafeNet Network HSM server.pem to client.



Note: At this point there are still no certificates in cert\server directory.

3. Use "vtl addserver" to register the SafeNet Network HSM with the client.
CAFile.pem is generated in the cert\server directory.

Cert\server directory after CAFile.pem is generated



Crystoki.ini after "vtl addserver"

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

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

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

[CardReader]
RemoteCommand=1
```

vtl verify results

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

The following SafeNet Network HSM Slots/Partitions were found:

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

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

Replace a SafeNet Network HSM Using the same IP

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

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



Note: “vtl verify” on client at this time would fail because the cert that the client has is for the old, removed SafeNet Network HSM.

3. Execute “vtl deleteserver -n <original IP>”

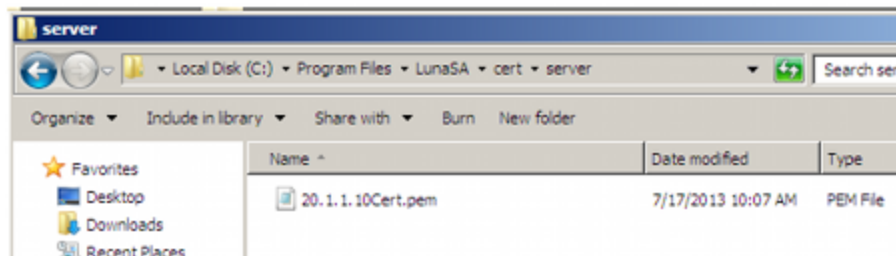
Deleting old SafeNet Network HSM from Client

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

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

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

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



4. Copy new server.pem to client

Copying new server.pem to client

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

5. Run vtl addserver using new server.pem

vtl addserver using new server.pem

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

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

6. Run vtl verify.

vtl verify results

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

The following SafeNet Network HSM Slots/Partitions were found:

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

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

Summary

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

Client side requirements review:

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

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

1. Note HA partition serial numbers

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

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

2. Run "lunacm ha -newGroup..."

A group is created with HA1 as Primary.

```
C:\Program Files\SafeNet\LunaClient>vtl haadmin -newGroup -label SomeHAGrp -serial 154702011
-password userpin
New group with label "SomeHAGrp" created at group number 1154702011.
Group configuration is:
```

```
HA Group label:  SomeHAGrp
HA Group Number: 1154702011
HA Group Slot #:  unknown
Synchronization: enabled
Group Members:   154702011
Standby Members: <none>
In Sync:         yes
```

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

Crystoki.ini after HA group is created

```
[Crystoki2]
LibNT=C:\Program Files\SafeNet\LunaClient\cryptoki.dll

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

```

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

```

```

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

```

```

[CardReader]
RemoteCommand=1

```

```

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

```

```

[HASynchronize]
SomeHAGrp=1

```

3. Add a secondary SafeNet Network HSM partition to the HA group with lunacm:> ha - addMember.

```

lunacm:> ha -addMember -group SomeHAGrp -serialNum 154702012 -password userpin
New group with label "SomeHAGrp" created at group number 1154702011.
Group configuration is:

```

```

      HA Group label:  SomeHAGrp
      HA Group Number: 1154702011
      HA Group Slot #:  6
      Synchronization: enabled
      Group Members:   154702011, 154702012
      Standby Members: <none>
      In Sync:         yes

```

Please use the command 'vtl haAdmin -synchronize' when you are ready to replicate data among all members of the HA group. (If you have additional members to add, you might wish to wait until you have added them before synchronizing to save time by avoiding multiple synchronizations.)

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

Crystoki.ini after second HA member is added

```

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

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

```

```
ClientPrivKeyFile=C:\Program Files\SafeNet\LunaClient\cert\client\20.1.1.20.pem
ServerName00=20.1.1.20
ServerPort00=1792
```

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

```
[CardReader]
RemoteCommand=1
```

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

```
[HASynchronize]
SomeHAGrp=1
```

Crystoki.ini after HA Only is enabled

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

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

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

```
[CardReader]
RemoteCommand=1
```

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

```
[HASynchronize]
SomeHAGrp=1
```

```
[HAConfiguration]
HAOnly=1
```

Crystoki.ini after "autorecovery" is enabled

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

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

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

[CardReader]
RemoteCommand=1

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

[HASynchronize]
SomeHAGrp=1

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

4. Show HA configuration results with vtl haAdmin -show

```
C:\Program Files\SafeNet\LunaClient>vtl haadmin -show
```

```
===== HA Global Configuration Settings =====
      HA Auto Recovery:  enabled
Maximum Auto Recovery Retry:  500
Auto Recovery Poll Interval:  60 seconds
      HA Logging:  disabled
Only Show HA Slots:  yes

===== HA Group and Member Information =====

      HA Group label:  SomeHAGrp
      HA Group Number: 1154702011
      HA Group Slot #: 1
      Synchronization: enabled
      Group Members:  154702011, 154702012
      Standby Members: <none>
```

Slot #	Member S/N	Member Label	Status
=====	=====	=====	=====
-	154702011	HA1	alive
-	154702012	HA2	alive

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

Replacing the Secondary HA Group Member

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

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

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

Frequently Asked Questions

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

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

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

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

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

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

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

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

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


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

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

HSM Initialization

This chapter describes how to initialize your HSM. It contains the following sections:

- "Initialization Overview for Password-Authenticated HSMs" below
- "Initialization Overview for PED-authenticated HSMs" on page 240
- "HSM Initialization and Zeroization" on page 243
- "Re-initialize an HSM" on page 244
- "Initialize an HSM With Existing Domain and Shared PED Keys" on page 244

Initialization Overview for Password-Authenticated HSMs

(This page is not instructions. This page is background information that might help make some operations more obvious.)

For SafeNet HSMs, there are two kinds of initialization:

- "hard" init - occurs when the HSM is in a factory [re]fresh state
- "soft" init - occurs when the HSM is not in factory [re]fresh state

Both are launched by the same command, `hsm init -l <hsmlabel>`.

Condition/Effect	Soft init	Hard init
SO authentication required?	Yes	No
Can set new HSM label	Yes	Yes
Creates new SO identity	No	Yes
Creates new Domain	No	Yes
Destroys partitions	Yes	No (none exist to destroy) *
Destroys SO objects	Yes	No (none exist to destroy) *

* `hsm factoryReset` was performed, and destroyed partitions and objects, before the hard init... otherwise, it could not be a hard init.

Hard Initialization

Coming from the factory, the SafeNet Network HSM:

- has network settings left over from our manufacturing process and not recommended for your production network
- has only default certificates in place

- has an undifferentiated HSM with no associations or ownership declared
- has not yet had virtual HSMs (HSM Partitions) created or assigned
- has not been introduced to the Clients (your Clients) with which it will be working.

Network setup of the appliance takes care of the first two items on that list. See "[Configure IP and Network Parameters](#)" on page 1 in the *Configuration Guide*.

Initialization takes care of the third item, which pertains specifically to the HSM portion of the appliance.

When you initialize a new (or factoryReset) HSM, several things happen, but the most important ones from your operational perspective are:

- you set up Security Officer or HSM Administrator (two names for the same entity) ownership of the HSM, and
- you apply a cloning domain to permit secure backup and restore, and secure cloning/replication of HSM objects to other HSMs.

For SafeNet Network HSM with Password Authentication, all authentication secrets, including the Security Officer authentication and the Cloning Domain secret are text strings that you type in at a keyboard (either via local serial console, or via SSH session).

From the `hsm init` command, the eventual outcome is an initialized HSM, that can be accessed by a specific SO password and Cloning Domain . How you get there can vary slightly, depending upon starting conditions. For this description, we assume a factory-fresh HSM. Alternatively, you can run `hsm factoryReset` (at the local serial console) to place the HSM in a similar "like new" state.

Initializing

- Issue the `hsm init` command, with a suitable HSM label - also include the HSM's SO password and a string for cloning domain (the domain determines with which other HSMs your HSM can clone objects).
- At this point, the HSM is initialized .
- Further actions are needed to prepare for use by your Clients, but you can now log in as SO/HSM Admin and perform HSM administrative actions.

Logging In, Once You Have Initialized

- To login, you issue the `hsm login` command at the `lunash:>` command line.
- When prompted, type the password. The HSM checks what it receives against what it expects. If it finds a mismatch, it records a bad-login attempt against the bad-login counter. You have two more chances to present the correct SO/HSM Admin authentication, or the SO is locked out and the HSM must be re-initialized (where it is zeroized and all contents are gone) before it can be used again.
- When you login successfully, the bad-login counter is reset to zero.

Soft Initialization

The above description covers the situation where your HSM is new from the factory, or where you have recently run `hsm factoryReset` command. The result of `hsm init` is different if the HSM is **not** in factory reset state.

If you run `hsm init -l <hsmlabel>` on an HSM that is currently in initialized state, then you are performing a "re-initialization", or a soft init, and not a full, hard initialization.

In this situation, `hsm init -label <hsmlabel>` means remove any partitions (and their contents) and erase any token objects that reside in SO space on the HSM. The SO identity is preserved, as is the cloning domain. The HSM label can be any string - you do not need to retain the previous label - you can change the previous label with this command. The password is required, to prove that you are entitled to perform the initialization.

Why choose Hard Init or Soft Init?

A good example of a situation where you might generally prefer to perform **soft initialization** is when provisioning with Crypto Command Center for virtual clients. When a client (virtual or otherwise) is done with a SafeNet HSM resource - say, a partition or a group of partitions - the resource must be cleared (removed and re-created, re-deployed) for the next customer.

Either kind of initialization operation takes care of destroying the partitions and contents, but a **soft init** leaves the SO identity and the cloning domain intact. The HSM remains within its established environment, under control of the Crypto Command Center administrator, who has no need to change SO and domain, but who does wish to create new user partitions for the next deployment.

A **hard initialization** (factoryreset followed by init) prepares the HSM for any environment, since the factoryreset removes any traces of the previous environment (SO and domain) and makes the HSM ready to accept new authentication.

The **hard init** is always the safest approach to take, since you can always choose to use the existing password and cloning-domain strings - emulating the end result of a soft init, but if you have security-policy reasons for not allowing the SO or domain to remain, the **hard init** addresses those reasons.

Similarly, if you had been validating an HSM in a laboratory environment before deploying it to a production HA environment, a **soft init** would leave the HSM with the lab domain in place. The domain used by your production HA group of HSMs would not match, thereby preventing the new HSM from cloning with that group (so no HA, no synchronization). A **hard init** of the new HSM when introducing it to the HA group would ensure that it was initialized with the domain needed to participate in that HA group.

Initialization Overview for PED-authenticated HSMs

For SafeNet HSMs, there are two kinds of initialization:

- "hard" init - occurs when the HSM is in a factory [re]fresh state
- "soft" init - occurs when the HSM is not in factory [re]fresh state

Both are launched by the same command, `hsm init -l <hsmlabel>`.

Condition/Effect	Soft init	Hard init
SO authentication required?	Yes	No
Can set new HSM label	Yes	Yes
Creates new SO identity	No	Yes
Creates new Domain	No	Yes
Destroys partitions	Yes	No (none exist to destroy) *
Destroys SO objects	Yes	No (none exist to destroy) *

* `hsm factoryReset` was performed, and destroyed partitions and objects, before the hard init... otherwise, it could not be a hard init.

Hard Initialization

Coming from the factory, the SafeNet HSM:

- has an undifferentiated HSM with no associations or ownership declared
- has not yet had virtual HSMs (HSM Partitions) created or assigned
- has not been introduced to the Clients (your Clients) with which it will be working.

Initialization takes care of the ownership by establishing roles and separations of authority.

When you initialize a new (or `factoryReset`) HSM, several things happen, but the most important ones from your operational perspective are:

- you set up Security Officer or HSM Administrator (two names for the same entity) ownership of the HSM, and
- you apply a cloning domain to permit secure backup and restore, and secure cloning/replication of HSM objects to other HSMs.

For SafeNet HSMs with PED (Trusted Path) Authentication, the HSM Security Officer authentication and the Cloning Domain secret are kept on portable physical memory devices called PED Keys (["About PED Keys" on page 288](#)). PED Keys can interact with the HSM via the SafeNet PED which provides a protected data path or "Trusted Path". Use of the Trusted Path for authentication prevents observation or interception of passwords such as you would type at a keyboard when seeking access to password authenticated HSMs.

From the `hsm init` command, the eventual outcome is an initialized HSM, that can be accessed by a specific blue SO PED Key and red Domain PED Key (or the Password authentication equivalents). How you get there can vary slightly, depending upon starting conditions. For this description, we assume a factory-fresh HSM and factory-fresh ('blank') PED Keys. Alternatively, you can run `hsm factoryReset` (at the local serial console) to place the HSM in a similar "like new" state.

Initializing

- Your SafeNet PED must be connected to the HSM, either locally/directly, or remotely via Remote PED connection (see ["About Remote PED" on page 370](#)), with "Awaiting command..." showing on its display.
- When you issue the `hsm init` command, the HSM passes control to the SafeNet PED, and the command line (`lunash:>`) directs you to attend to the PED prompts.
- A "default" login is preformed, just to get started (you don't need to supply any authentication for this step).
- SafeNet PED asks: "Do you wish to reuse an existing keyset?". If the answer is NO, the HSM creates a new secret which will reside on both the HSM and the key (or keys) that is (or are) about to be imprinted. If the answer is YES, then the HSM does not create a new secret and instead waits for one to be presented via the PED.
- SafeNet PED requests a blue PED Key. It could be blank to begin, or it could have a valid secret from another HSM (a secret that you wish to preserve), or it could have a secret that is no longer useful.
- SafeNet PED checks the key you provide. If the PED Key is not blank, and your answer to "...reuse an existing keyset" was "Yes", then SafeNet PED proceeds to copy the secret from the PED Key to the HSM.
- If the key is not blank, and your answer to "...reuse an existing keyset" was "No", then the PED inquires if you wish to overwrite its contents with a new HSM secret. If the current content of the key is of no value, you say "Yes". If the current content of the key is a valid secret from another HSM (or if you did not expect the key to hold any data) you can remove it from the PED and replace it with a blank key or a key containing non-useful data, before you

answer "Yes" to the 'overwrite' question. Even so, SafeNet PED asks "Are you sure...".

- Assuming that you are using a new secret, and not reusing an existing one, SafeNet PED asks if you wish to split the new HSM secret. It does this by asking for values of "M" and "N". You set those values to "1" and "1" respectively, unless you require MofN split-secret, multi-person access control for your HSM (See ["Using MofN" on page 343](#) for details).
- SafeNet PED asks if you wish to use a PED PIN (an additional secret; see ["What is a PED PIN?" on page 296](#) for more info).
- If you just press ENTER (effectively saying 'no' to the PED PIN option), then the secret generated by the HSM is imprinted on the PED Key, that same secret is retained as-is on the HSM, and the same secret becomes the piece needed to unlock the Security Officer/HSM Admin account on the HSM.
- If you press some digits on the PED keypad (saying 'yes' to the PED PIN option), then the PED combines the HSM-generated secret with your PED PIN and feeds the combined data blob to the HSM. The HSM throws away the original secret and takes on the new, combined secret as its SO/HSM Admin secret.
- The PED Key contains the original HSM-generated secret, but also contains the flag that tells the PED whether to demand a PED PIN (which is either no digits, or a set of digits that you supplied, and must supply at all future uses of that PED Key).
- SafeNet PED gives you the option to create some duplicates of this imprinted key. You should make at least one duplicate for backup purposes. Make additional duplicates if your security policy permits, and your procedures require them.
- Next, SafeNet PED requests a red Domain PED Key. The HSM provides a cloning Domain secret and the PED gives you the option to imprint the secret from the HSM, or to use a domain that might already be on the key. You choose appropriately. If you are imprinting a new Domain secret, you have the same opportunities to split the secret, and to apply a PED PIN "modifier" to the secret. Again, you are given the option to create duplicates of the key.
- At this point, the HSM is initialized and SafeNet PED passes control back to the appliance (lunash:>).
- Further actions are needed to prepare for use by your Clients, but you can now log in as SO/HSM Admin and perform HSM administrative actions.

Logging In, Once You Have Initialized

- To login, you issue the role login command at the luncm:> command line. Control is passed to SafeNet PED.
- The PED prompts for the blue SO PED Key. You insert that PED Key.
- If there was no PED PIN (you chose none at initialization time), then the PED combines the secret on the blue key with ... nothing... and the unchanged secret is passed to the HSM. The HSM recognizes the secret and logs you in. This assumes that you provided the correct blue PED Key.
- If there was a PED PIN (you added one at initialization time), then you type it on the PED keypad, SafeNet PED combines the secret from the key with the digits that you type, and the modified secret is passed to the HSM. The HSM recognizes the modified secret and logs you in. This assumes that you provided the correct blue PED Key **and** the correct PED PIN digits.
- If you type an incorrect PED PIN, what the HSM receives is much the same as if you presented a wrong PED Key. The HSM checks what it receives against what it expects, finds a mismatch and records a bad-login attempt against the bad-login counter. You have two more chances to present the correct SO authentication, or the SO is locked out and the HSM must be re-initialized (where it is zeroized and all contents are gone) before it can be used again.
- When you login successfully, the bad-login counter is reset to zero.

If you had also elected to split the login secret when you initialized, then the above sequence would need quantity M different blue keys from the set of quantity N, in order to reconstruct the needed secret (along with PED PINs, or not, for each partial-secret blue key).

Soft Initialization

The above description covers the situation where your HSM is new from the factory, or where you have recently run `hsm factoryReset` command. The result of `hsm init` is different if the HSM is **not** in factory reset state.

If you run `hsm init -l <hsmlabel>` on an HSM that is currently in initialized state, then you are performing a "re-initialization", or a soft init, and not a full, hard initialization.

In this situation, `hsm init -label <hsmlabel>` means remove any partitions (and their contents) and erase any token objects that reside in SO space on the HSM. The SO identity is preserved, as is the cloning domain. The HSM label can be any string - you do not need to retain the previous label - you can change the previous label with this command. You are prompted by SafeNet PED to insert the currently-valid blue SO PED Key (it is not changed by a soft init; it is needed only to validate your right to perform the soft initialization) and press [Enter] on the PED keypad. No other interaction is needed.

Why choose Hard Init or Soft Init?

A good example of a situation where you might generally prefer to perform **soft initialization** is when provisioning with Crypto Command Center for virtual clients. When a client (virtual or otherwise) is done with a SafeNet HSM resource - say, a partition or a group of partitions - the resource must be cleared (removed and re-created, re-deployed) for the next customer.

Either kind of initialization operation takes care of destroying the partitions and contents, but a **soft init** leaves the SO identity and the cloning domain intact. The HSM remains within its established environment, under control of the Crypto Command Center administrator, who has no need to change SO and domain, but who does wish to create new user partitions for the next deployment.

A **hard initialization** (factoryreset followed by init) prepares the HSM for any environment, since the factoryreset removes any traces of the previous environment (SO and domain) and makes the HSM ready to accept (or generate) new blue and red key data.

The **hard init** is always the safest approach to take, since you can always choose to use the existing blue and red PED Keys and imprint those onto the factoryreset HSM - emulating the end result of a soft init, but if you have security-policy reasons for not allowing the SO or domain to remain, the **hard init** addresses those reasons.

Similarly, if you had been validating an HSM in a laboratory environment before deploying it to a production HA environment, a **soft init** would leave the HSM with the lab domain in place (in a soft init, the PED does not prompt for insertion of the red PED Key, since the HSM already has a domain). The domain used by your production HA group of HSMs would not match, thereby preventing the new HSM from cloning with that group (so no HA, no synchronization). A **hard init** of the new HSM when introducing it to the HA group would ensure that it was initialized with the domain needed to participate in that HA group.

HSM Initialization and Zeroization

Ideally, the `hsm init` command is used once, when you first configure your SafeNet 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 SafeNet system into a "live" or "production" environment.

For further detail and for explanations of the concepts "hard" init and "soft" init, see ["Initialization Overview for PED-authenticated HSMs" on page 240](#) and ["Initialization Overview for Password-Authenticated HSMs" on page 238](#).

Additional Notes

The SafeNet shell command `hsm factoryReset` puts the HSM in a zeroized state. (See ["What Does Zeroized Mean?" on page 404](#).) 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 the command is accepted only via the local serial console. It is assumed that you provide sufficient physical security for your HSM appliance(s). An attacker who could interrupt or deny your use of the HSM by gaining access to your premises to make a serial connection and issue destructive commands could as easily steal or physically destroy the HSM while in your server room.

If you are taking a SafeNet Network HSM out of service, to go into storage, or to be shipped to another location (or back to SafeNet), then after you perform `hsm factoryReset`, perform `hsm init` to overwrite any labels or settings that you previously made.

View a table that compares and contrasts various "deny access" events or actions that are sometimes confused. ["Comparison of Destruction/Denial Actions" on page 400](#)

Re-initialize an HSM

To initialize (see ["Initializing a PED-Authenticated HSM" on page 1](#)) in the *Configuration Guide* or to re-initialize an HSM, use the command:

```
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 SafeNet HSM. However, if you have made major changes in your system/deployment, or if you are moving a SafeNet 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: Also, some HSM policy changes are destructive of HSM contents (a security measure), and require re-initialization before you can continue to use the HSM. In the case where you intend to make a destructive HSM policy change, be sure to back up any important objects and keys so that they can be restored after the policy change and subsequent re-initialization.

Initialize an HSM With Existing Domain and Shared PED Keys

For two SafeNet HSMs, the following procedure assumes that you wish to have a set of PED Keys that will work with either HSM. One HSM is already initialized, so you have a full set of PED Keys, imprinted with the authentication data and the domain for that HSM. You want the second HSM to share the same domain (for backup, and the ability to

restore to either HSM from a Backup token), and both the old and the new PED Keys should work interchangeably with both HSMs.

For this example procedure, HSMs are designated:

- HSM 1, with its PED Keys Blue K1 and Red K1, and
 - HSM 2, with its PED Keys Blue K2 and Red K2.
1. Ensure that you can log in to HSM 1 as a Security Officer using Blue K1 (if not, then do not continue with the procedure).
 2. Log out.
 3. Begin initialization of HSM 2.
 4. Insert Blue K1 at the PED prompt, and when asked if you would "like to reuse an existing keyset", answer [YES] on the PED keypad.
 5. Duplicate Blue K1 to Blue K2 when prompted. (That is, when asked "Are you duplicating this keyset", answer [YES], then insert the target Blue K2).
 6. When "Generating a domain" appears, insert Red K1 at the prompt.
When asked "Would you like to reuse an existing keyset", answer [YES].
 7. Duplicate Red K1 to Red K2.

The procedure to make a backup of the black PED Key (for HSM Partitions) would be similar to the procedure for the blue PED Key.



Note: You might receive a message that the key is blank, or that it contains valid data (for whatever type of key it was previously) and asking if you wish to overwrite. If the PED has indicated that the target PED Key is occupied and you are not certain that any authentication it contains is obsolete, then you should not allow it to be overwritten. Either remove the current, problematic key, insert another "blank" target key, and press [ENTER], or abort the operation. To abort, remove the PED Key and wait for PED time-out. Do NOT press [ENTER] at the "overwritten" message, if that is not your intent. Retry when you have sorted out your PED Keys and are confident that your target key is blank or contains truly obsolete authentication that can legitimately be overwritten.

If you wish to have a separate set of keys for each HSM, then instead of following the procedure as written you should use the Blue K2 and Black K2 for HSM2 and answer 'NO' to the question "Would you like to reuse an existing keyset?" This will imprint/overwrite the new blue or black keys making them specific to HSM2. For the Red key you should still insert Red K1 and answer 'YES' to the "Would you like to reuse an existing keyset?" question (the token/HSMs must share a common domain, or backup/restore cannot take place).

HSM Partitions

This chapter describes how to administer HSM administrative and application partitions on the HSM. It contains the following sections:

- "HSM Partitions" below
- "Partition Creation - Notes" on page 1
- "Partition Creation with Policy Template Using LunaCM" on page 252
- "Partition Creation with Policy Template Using Lunash" on page 1
- "Separation of HSM Workspaces" on page 263
- "Configured and Registered Client Using an HSM Partition" on page 267
- "About Activation and Auto-Activation " on page 268
- "Removing Partitions" on page 271
- "Security of Your Partition Challenge" on page 271
- "Frequently Asked Questions" on page 272

HSM Partitions

HSM Partitions are independent logical HSMs that reside within the SafeNet HSM inside, or attached to, your host computer or appliance. Each HSM Partition has its own data, access controls, security policies, and separate administration access for at least some roles, independent from other HSM partitions (if your HSM supports more than one). Depending on the product, the HSM can contain multiple HSM partitions, and each partition can be associated with one or more Clients. Each HSM Partition has a special administrative account or role, who manages it.

HSMs with firmware older than version 6.22.0 had three types of partitions:

- the HSM administrative partition, administered by the HSM SO
- the Auditor partition, accessible and administered by the HSM Auditor role, only
- application partition(s) administered at a high level by the HSM SO, but administered and operated at an operational level by the User or Crypto Officer role

HSMs with firmware 6.22.0 or newer can have three types of partitions:

- the HSM administrative partition, administered by the HSM SO (see Note below)
- legacy-style application partition(s) administered at a high level by the HSM SO, but administered and operated at an operational level by the User or Crypto Officer role (with optional Crypto User)

or

- PPSO application partitions (requires that the PPSO capability is installed) that are created by the HSM SO, but are thereafter owned by their own local SOs, and administered and operated at an operational level by the Crypto

Officer role (with optional Crypto User) - installing the PPSO capability is destructive, and requires that you re-create partitions, if you already had any.

Note: For HSMs with firmware 6.22.0 or newer, the Auditor role does not have an independent partition, but controls an area within the HSM administrative partition. The role and its objects are not seen or touched by the HSM SO.



Operationally, there is no difference from previous releases. The only caveat is that if you update an older HSM to firmware 6.22.0 or newer, the old Audit logging stops and you must initialize the Audit user again, and configure audit logging. It is perfectly acceptable to re-use the Auditor credentials (white PED Key).

HSM Partitions can be thought of as 'safe deposit boxes' that reside within the K6 Cryptographic Engine's 'vault'. The vault itself offers an extremely high level of security for all the contents inside; additionally, each safe deposit box also has its own security and access controls; while the bank managers might have access to the vault, they still cannot open the individual safe deposit boxes, because only the owner of the safe deposit box holds the key that opens it.

A legacy application partition was/is owned by the HSM SO, who assigns a User or Crypto Officer to handle day-to-day management of partition contents, creation, use, and destruction of keys and objects, and so on. PPSO application partitions (where HSM firmware is version 6.22.0 or newer, and the PPSO capability is applied) have their own partition SO, distinct from the HSM SO. The HSM SO initializes the HSM, sets HSM-wide policies, creates an empty application partition, and hands off complete control to whomever is to become the partition SO. Thereafter, the HSM has no oversight and can do nothing with the partition except to delete it, if that is ever required. The Partition SO then initializes the partition creating a Crypto Officer

Note: If you are both

- upgrading from an earlier firmware version to HSM firmware 6.22.0 (or newer)

AND

- applying the Per-Partition SO (PPSO) capability update,

be aware that the PPSO capability update is destructive. Therefore, there is no need to re-size partitions.



Instead, to avoid unnecessary duplication of effort, you should

- safeguard (archive) any existing partition contents,
- then zeroize the HSM for a clean update,
- then perform both the firmware AND capability updates,
- and finally restore to new partitions.

Compare Legacy Partition vs PPSO Partition

This page compares the implications and handling of legacy partitions and Per-Partition SO partitions.

Historically, SafeNet HSMs had a single Security Officer/HSM Administrator (the SO) for the base HSM and for any partitions on that HSM. "Per-Partition SO" (abbreviated PPSO) is an setting that can optionally be applied to HSM partitions to more completely separate the roles. PPSO allows creation of HSM Partitions that individually have a Security Officer administrative role, in addition to the Crypto Officer and Crypto User roles. This makes each such partition behave more like a stand-alone HSM, and is a major step toward full multi-tenancy in cloud environments.

The possible application partitions that can be viewed and manipulated in the current release include:

- Legacy partitions, on HSM with firmware older than version 6.22.0
- Legacy-style partitions, on HSM with firmware 6.22.0 or newer, but partition is created to have the HSM SO as the partition SO (no PSO)
- PPSO partitions, on HSM with firmware 6.22.0 or newer, and partition is created uninitialized, ready to have its own partition SO (PSO)

Action or Feature	Legacy HSM (f/w pre-6.22.0)	Current HSM (f/w 6.22.0 or newer)	
		Legacy Partitions	PPSO
Initialize the HSM	Run hsm init command	Run hsm init command	Run hsm init command
Number of Partitions	Legacy = number of application partitions (SO space was special case)	Current = HSM SO partition + number of application partitions	Current = HSM SO partition + number of application partitions
Create Partitions	HSM SO creates with partition create - partition <partitionname> command	HSM SO creates with partition create - partition <partitionname> command	HSM SO creates with partition create - haspso -partition <partitionname> command
Status of newly created partition	Has (at least) User or Crypto Officer role and has cloning domain	Has (at least) User or Crypto Officer role and has cloning domain	Partition framework exists as "factory reset" token, with no roles and no domain
Next action after partition create	give password to client, who/which starts doing crypto operations on partition	give password to client, who/which starts doing crypto operations on partition	with slot selected, partition init to create application partition SO and domain (no role exists prior to this action, so no authentication is required to create the SO in a freshly-created partition),
Next action after application partition SO is created with role init	[skip - not applicable]	[skip - not applicable]	application partition SO logs in and creates CO with role init
Next action after SO does with role init to create CO	[skip - not applicable]	[skip - not applicable]	CO logs in and creates CU with role init
Next action after creation of Crypto Officer and Crypto User	(already done, above)	(already done, above)	give password to client, who/which starts doing crypto operations on partition
Partition administrative	HSM Security Officer (SO) (blue PED Key for PED-	HSM Security Officer (SO) (blue PED Key for PED-	application partition Security Officer (PSO)

Action or Feature	Legacy HSM (f/w pre-6.22.0)	Current HSM (f/w 6.22.0 or newer)	
		Legacy Partitions	PPSO
credential	authenticated HSM)	authenticated HSM)	(blue PED Key for PED-authenticated HSM)
Crypto Officer (read-write partition user) primary credential	- CO password for PW-authenticated - black Partition Owner PED Key for PED-authenticated	- CO password for PW-authenticated - black Partition Owner PED Key for PED-authenticated	- CO password for PW-authenticated - black Crypto Officer PED Key for PED-authenticated
Crypto User read-only primary credential	- CU password for PW-authenticated - black Partition Owner PED Key for PED-authenticated	- CU password for PW-authenticated - black Partition Owner PED Key for PED-authenticated	- CU password for PW-authenticated - gray Crypto User PED Key for PED-authenticated
Crypto Officer secondary credential	- CO password for PW-authenticated (there is only one CO credential for PW-auth) - CO challenge secret (a.k.a. password) for PED-authenticated Used by application to create/modify crypto objects	- CO password for PW-authenticated (there is only one CO credential for PW-auth) - CO challenge secret (a.k.a. password) for PED-authenticated Used by application to create/modify crypto objects	- CO password for PW-authenticated (there is only one CO credential for PW-auth) - CO challenge secret (a.k.a. password) for PED-authenticated Used by application to create/modify crypto objects
Crypto User secondary credential	- CU password for PW-authenticated (there is only one CU credential for PW-auth) - CU challenge secret (a.k.a. password) for PED-authenticated Used by application to use (but not modify) crypto objects	- CU password for PW-authenticated (there is only one CU credential for PW-auth) - CU challenge secret (a.k.a. password) for PED-authenticated Used by application to use (but not modify) crypto objects	- CU password for PW-authenticated (there is only one CU credential for PW-auth) - CU challenge secret (a.k.a. password) for PED-authenticated Used by application to use (but not modify) crypto objects
Activate Partitions (PED-auth only)	HSM SO sets policy to enable Activation (optionally also to enable AutoActivation if HSM is part of an HSM appliance). Crypto Officer activates with partition activate command.	HSM SO sets policy to enable Activation; Crypto Officer or Crypto User logs in with role login	Partition SO sets policy to enable Activation; Crypto Officer or Crypto User logs in with role login
Deactivate Partitions	partition deactivate command	partition deactivate command	partition deactivate command

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

Authentication in General

The HSM and Partition commands require the appropriate HSM SO and Partition CO or CU passwords for password-authenticated HSMs, or blue and black and gray PED Keys for PED-authenticated HSMs. See "PED Keys and Operational Roles" on page 322.

When a command is issued to the HSM that requires HSM or application Partition authentication, the HSM checks for logged-in status and either continues with the operation or issues an error message that the required user is not logged in. When performing the required login operation, a password-authenticated HSM looks for the appropriate password. A PED-authenticated HSM looks to the connected SafeNet PED. The PED responds by prompting you for actions involving the appropriate PED Keys and the PED keypad. If the PED gets the appropriate response, it confirms the authentication back to the HSM, via the PED interface (the Trusted Path). The required PED Keys would be:

- the blue key(s) needed when the HSM Admin logs in, or issues an `hsm` command.
- the black key(s), 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 SafeNet PED when you perform administrative operations.

A gray PED Key is just a "black" PED Key with a visible label to differentiate the Crypto User from the Crypto Officer. The PED treats them identically, and prompts only for "black" PED Keys. It is up to you to keep track of who is authenticating - the gray label is meant only to make that easier when handling physical PED Keys. If you prefer (or if you are using pre-existing key-sets), you could have one black PED Key for Crypto Officer, and another black PED Key for Crypto User.

The HSM authentication for PED-authenticated HSMs can consist of:

- presenting the required PED Key(s) and pressing [ENTER] on the keypad, 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 HSM will carry out HSM or Partition commands (according to the level of authentication that you invoked).

Authentication and Access Control for Clients

However, the point of the HSM appliance is that authorized remote Client applications must be able to access their Partitions, in order to perform useful work (such as signing, verifying, encrypting, decrypting), and also that unauthorized clients be prevented from doing so. Before authorized access can happen, the Partition must be in a logged-in state (as described above) by means of the black PED Key.

To preclude access by unauthorized client applications, the HSM requires that two authentication conditions be in place:

- The Partition must be readied to accept Client access in a login state authenticated by the black PED Key which is accepted only via the PED (this gives administrative access to the Partition, and opens the Partition to Client access, but only if the next authentication element is supplied),
- The Client must provide its credentials in the form of an authentication secret (a text-string password).

The Client authentication is the Partition Password that was displayed by the PED, and recorded by you, at the time the Partition was created (or it is the string to which you changed that original Partition Password, for your convenience, or to fit your security scheme).

If you provide that Partition Password only to registered, authorized Clients, and if they in turn keep it secret, then no unauthorized client can ever access the HSM appliance or its HSM. If you place an HSM Partition into a login state, then any registered application that presents the Partition Password is welcomed as an authorized Client.

The login state continues as long as a Client has the connection open to the Partition.

For a password-authenticated HSM, there is only the one secret for the CO, or the one secret for the CU. The client application is given exactly the same partition password that the CO uses, or exactly the same partition password that the CU uses, for administrative purposes, but the client uses it for operational cryptographic purposes.

Activation (PED-auth only)

Activation is just a login with explicit caching of the Partition login data, on the HSM. This is convenient so that you can remove the black PED key (perhaps to allow other uses of the PED, such as administrative logins by the HSM Admin), while ensuring that access by Clients is not stopped, and that nobody is required to be present to press [ENTER] on the keypad for the benefit of Client applications.

To use Activation, you must first allow it by setting Partition Policy 22 (Allow Activation) to *on*, for each Partition that you create. If the Policy (22, Allow Activation) is on, then the Partition Owner (or Crypto Officer) can activate the partition for use. This is done differently depending upon the HSM firmware.

- for legacy partitions, or any partition on an HSM with firmware older than version 6.22.0, issue the **partition activate** command
- for PPSO partitions, or any partition on an HSM with firmware version 6.22.0 or newer, log in as Crypto Officer or Crypto User with the **role login** command.

SafeNet PED prompts for the black PED Key(s) and PED PIN if appropriate.

Once you provide it, the 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 HSM.

With the application partition activated, you can remove the black PED Key and keep it in your pocket or in safe storage. Activation remains on, and any Client with the Partition Password (also known as "challenge secret") is able to connect and perform operations on the Partition.

Activation is a modest advantage for Clients that connect and maintain an open session. It is an indispensable advantage in cases where Clients repeatedly connect or open a session to perform a task and then disconnect or close the session.

AutoActivation (PED-auth only)

AutoActivation allows automatic re-activation of the Partition, using the cached Partition-Owner/Crypto-Officer authentication data, in the event of a restart or a short power outage (up to 2 hours). That is, the Activated state can recover to allow Clients to re-connect and continue using the Partition, without need for human intervention to insert the black PED Key and press [ENTER] on the PED keypad.

AutoActivation, which you set by the **partition changePolicy** command, requires that Partition Policy 23 (Allow AutoActivation) be *on*, for the affected Partition.

If the authentication data requires refreshing, then the PED prompts you to insert the appropriate black PED Key and press [ENTER] - that is, a black PED Key that was imprinted with the Partition authentication data for the particular Partition. Once control returns to the command line, and a message announces success, you can remove the black PED Key and store it away. Client applications can begin using the HSM Partition.

We anticipate that most customers will set Partition Policy 23 Allow auto-activation (battery-backed caching of partition authentication) to "On" for their partitions, to ensure the convenience (up-time) of their clients.

Customers who prefer to not set auto-activation On, but who keep their SafeNet HSMs located remotely from their administrative staff, might prefer to 'manually' resume partition activation by means of Remote PED. These options are entirely a matter of your preference and of your security policy.

Partition Creation with Policy Template Using LunaCM

Partition policy templates are a convenience feature to create application partitions with predetermined, non-default, partition policy settings, allowing more efficient provisioning of multiple application partitions having identical (or similar) policy settings.

Policy Templates are named combinations of partition policy settings that you can apply to an application partition at partition-creation time. If no policy template exists, or if you choose not to apply a policy template when you create a partition, then the partition comes into existence with the HSM's default policy set.

You can name one policy template when you create a partition, and that one is applied if it exists. You cannot apply more than one to a partition. You cannot "overwrite" a policy value set by applying a different partition policy template to an existing application partition after the partition has been created.

You always have the option to change partition policies manually, individually, at any time, in the current slot/partition. If a policy template was not applied, then any changes you make with individual **partition changePolicy** commands are enacted upon existing policies that reflect the original default policy values of the HSM. If a policy template was applied, then any changes you make with individual **partition changePolicy** commands are enacted upon a set of policies that reflect the original default policy values modified by the changes recorded in the template at the time that template was applied (partition creation).

The examples on this page apply to manipulating application partitions via lunacm.

Process for a New Template

The general procedure is as follows:

- Create (and load for editing) a new, unnamed partition policy template. The possible policy codes, along with their default settings, are displayed.
- Make changes to those default values, one at a time, until you are satisfied. Each change is echoed back.
- Save the new partition policy template, applying a name that is unique and easily recognized, and also applying additional descriptive text to assist yourself and future users to recall the purpose of this specific template among any others you might create.
- Create an application partition, specifying a particular partition policy template by name. This creates the partition with policies applied to it, conforming to the selected template, different from the default set for the HSM.

Create and apply a new partition policy template

For this example, before starting, here are the policy values for a default partition that was created without using a template:


```
lunacm:> partition showpolicies
```

Partition Capabilities

```

0: Enable private key cloning : 1
1: Enable private key wrapping : 0
2: Enable private key unwrapping : 1
3: Enable private key masking : 0
4: Enable secret key cloning : 1
5: Enable secret key wrapping : 1
6: Enable secret key unwrapping : 1
7: Enable secret key masking : 0
10: Enable multipurpose keys : 1
11: Enable changing key attributes : 1
15: Allow failed challenge responses : 1
16: Enable operation without RSA blinding : 1
17: Enable signing with non-local keys : 1
18: Enable raw RSA operations : 1
20: Max failed user logins allowed : 10
21: Enable high availability recovery : 1
22: Enable activation : 1
23: Enable auto-activation : 1
25: Minimum pin length (inverted: 255 - min) : 248
26: Maximum pin length : 255
28: Enable Key Management Functions : 1
29: Enable RSA signing without confirmation : 1
30: Enable Remote Authentication : 1
31: Enable private key unmasking : 1
32: Enable secret key unmasking : 1
33: Enable RSA PKCS mechanism : 1
34: Enable CBC-PAD (un)wrap keys of any size : 1
35: Enable private key SFF backup/restore : 1
36: Enable secret key SFF backup/restore : 1
37: Enable Secure Trusted Channel : 1

```

Partition Policies

```

0: Allow private key cloning : 1
1: Allow private key wrapping : 0
2: Allow private key unwrapping : 1
3: Allow private key masking : 0
4: Allow secret key cloning : 1
5: Allow secret key wrapping : 1
6: Allow secret key unwrapping : 1
7: Allow secret key masking : 0
10: Allow multipurpose keys : 1
11: Allow changing key attributes : 1
15: Ignore failed challenge responses : 1
16: Operate without RSA blinding : 1
17: Allow signing with non-local keys : 1
18: Allow raw RSA operations : 1
20: Max failed user logins allowed : 10
21: Allow high availability recovery : 1
22: Allow activation : 0
23: Allow auto-activation : 0
25: Minimum pin length (inverted: 255 - min) : 248
26: Maximum pin length : 255
28: Allow Key Management Functions : 1
29: Perform RSA signing without confirmation : 1
30: Allow Remote Authentication : 1

```

```

31: Allow private key unmasking : 1
32: Allow secret key unmasking : 1
33: Allow RSA PKCS mechanism : 1
34: Allow CBC-PAD (un)wrap keys of any size : 1
35: Allow private key SFF backup/restore : 1
36: Allow secret key SFF backup/restore : 1
37: Force Secure Trusted Channel : 0

```

Command Result : No Error

Now, create a partition policy template and then create a new application partition using the new template.



Note: You must be in the administrative (HSM SO) slot in order to create a partition policy template.

1. Use command **partition policyTemplateCreate** to create a new partition policy template:

```
lunacm:> partition policytemplatecreate
```

Code	Description	Destructive		
		Value	Off-To-On	On-To-Off
0	Allow private key cloning	On	Yes	No
1	Allow private key wrapping	Off	Yes	No
2	Allow private key unwrapping	On	No	No
3	Allow private key masking	Off	Yes	No
4	Allow secret key cloning	On	Yes	No
5	Allow secret key wrapping	On	Yes	No
6	Allow secret key unwrapping	On	No	No
7	Allow secret key masking	Off	Yes	No
10	Allow multipurpose keys	On	Yes	No
11	Allow changing key attributes	On	Yes	No
15	Ignore failed challenge responses	On	Yes	No
16	Operate without RSA blinding	On	Yes	No
17	Allow signing with non-local keys	On	No	No
18	Allow raw RSA operations	On	Yes	No
20	Max failed user logins allowed	10	N/A	N/A
21	Allow high availability recovery	On	No	No
22	Allow activation	On	No	No
23	Allow auto-activation	On	No	No
24	Allow indirect login	Off	No	No
25	Minimum pin length (inverted: 255 - min)	248	N/A	N/A
26	Maximum pin length	255	N/A	N/A
28	Allow Key Management Functions	On	Yes	No
29	Perform RSA signing without confirmation	On	Yes	No
30	Allow Remote Authentication	On	No	No
31	Allow private key unmasking	On	No	No
32	Allow secret key unmasking	On	No	No
33	Allow RSA PKCS mechanism	On	Yes	No
34	Allow CBC-PAD (un)wrap keys of any size	On	Yes	No
35	Allow private key SFF backup/restore	Off	Yes	No

36	Allow secret key SFF backup/restore	Off	Yes	No
37	Force Secure Trusted Channel	Off	No	Yes

```
Type 'proceed' to continue, or 'quit'
to quit now.
> proceed
```

Successfully created and loaded the new partition policy template.

Use 'partition policyTemplateChange' to edit the template and 'partition policyTemplateSave' to save the template once you have applied all necessary changes.

Command Result : No Error

2. Use command **partition policyTemplateChange** to change some policy values in the new partition policy template:

```
lunacm:> partition policyTemplateChange -policy 25 -value 246
```

Code	Description	Destructive		
		Value	Off-To-On	On-To-Off
25	Minimum pin length (inverted: 255 - min)	246	N/A	N/A

Command Result : No Error

```
lunacm:> partition policyTemplateChange -policy 20 -value 9
```

Code	Description	Destructive		
		Value	Off-To-On	On-To-Off
20	Max failed user logins allowed	9	N/A	N/A

Command Result : No Error

```
lunacm:> partition policyTemplateChange -policy 7 -on non-destructive
```

Code	Description	Destructive		
		Value	Off-To-On	On-To-Off
7	Allow secret key masking	Off	No	No

Command Result : No Error

3. Use command **partition policyTemplateSave** to save the new partition policy template with its modified policy values:

```
lunacm:> partition policyTemplateSave -name sample01
```

sample01 successfully saved.

Command Result : No Error

```
lunacm:> partition policyTemplateList
```

Name	Description
sample01	

sample01

No partition policy template is currently loaded.

Command Result : No Error

4. Use command **partition create** with the **-policytemplate** option to create a new application partition, using the partition policy template that you previously created:

```
lunacm:> partition create -label parfortemplate -policyTemplate sample01
```

Please attend to the PED.

Command Result : No Error

```
lunacm:> slot set slot 0
```

Current Slot Id: 0 (Luna User Slot 6.24.0 (PED) Signing With Cloning Mode)

Command Result : No Error

```
lunacm:> partition showpolicies
```

Partition Capabilities

```

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

```

```
37: Enable Secure Trusted Channel : 1
```

Partition Policies

```
0: Allow private key cloning : 1
1: Allow private key wrapping : 0
2: Allow private key unwrapping : 1
3: Allow private key masking : 0
4: Allow secret key cloning : 1
5: Allow secret key wrapping : 1
6: Allow secret key unwrapping : 1
7: Allow secret key masking : 0
10: Allow multipurpose keys : 1
11: Allow changing key attributes : 1
15: Ignore failed challenge responses : 1
16: Operate without RSA blinding : 1
17: Allow signing with non-local keys : 1
18: Allow raw RSA operations : 1
20: Max failed user logins allowed : 9
21: Allow high availability recovery : 1
22: Allow activation : 0
23: Allow auto-activation : 0
25: Minimum pin length (inverted: 255 - min) : 246
26: Maximum pin length : 255
28: Allow Key Management Functions : 1
29: Perform RSA signing without confirmation : 1
30: Allow Remote Authentication : 1
31: Allow private key unmasking : 1
32: Allow secret key unmasking : 1
33: Allow RSA PKCS mechanism : 1
34: Allow CBC-PAD (un)wrap keys of any size : 1
35: Allow private key SFF backup/restore : 1
36: Allow secret key SFF backup/restore : 1
37: Force Secure Trusted Channel : 0
```

Command Result : No Error

Modify a partition template, then apply the modified partition template

For this example, we create an application using a partition template that has only one policy modified, then change the template to modify an additional policy, and create yet another partition to which we apply the modified partition template:



Note: You must be in the administrative (HSM SO) slot in order to create, load, and modify a partition policy template.

1. Create and save partition policy template Sample02 with policy 22 set to On, but policy 23 not set (see previous example for steps).
2. Use command **partition create** with the **-policytemplate** option to create a new application partition, using partition policy template Sample02 previously created:

```
lunacm:> partition create -label parfortemplateagain -policyTemplate Sample02
```

Please attend to the PED.

Command Result : No Error

3. Change to the slot of the newly-created partition and use command **partition showpolicies** to show the policies of the new partition:

```
lunacm:> slot set slot 0
```

```
Current Slot Id:      0      (Luna User Slot 6.24.0 (PED) Signing With Cloning Mode)
```

Command Result : No Error

```
lunacm:> partition showpolicies
```

Partition Capabilities

```

0: Enable private key cloning : 1
1: Enable private key wrapping : 0
2: Enable private key unwrapping : 1
3: Enable private key masking : 0
4: Enable secret key cloning : 1
5: Enable secret key wrapping : 1
6: Enable secret key unwrapping : 1
7: Enable secret key masking : 0
10: Enable multipurpose keys : 1
11: Enable changing key attributes : 1
15: Allow failed challenge responses : 1
16: Enable operation without RSA blinding : 1
17: Enable signing with non-local keys : 1
18: Enable raw RSA operations : 1
20: Max failed user logins allowed : 10
21: Enable high availability recovery : 1
22: Enable activation : 1
23: Enable auto-activation : 1
25: Minimum pin length (inverted: 255 - min) : 248
26: Maximum pin length : 255
28: Enable Key Management Functions : 1
29: Enable RSA signing without confirmation : 1
30: Enable Remote Authentication : 1
31: Enable private key unmasking : 1
32: Enable secret key unmasking : 1
33: Enable RSA PKCS mechanism : 1
34: Enable CBC-PAD (un)wrap keys of any size : 1
35: Enable private key SFF backup/restore : 1
36: Enable secret key SFF backup/restore : 1
37: Enable Secure Trusted Channel : 1
```

Partition Policies

```

0: Allow private key cloning : 1
1: Allow private key wrapping : 0
2: Allow private key unwrapping : 1
3: Allow private key masking : 0
4: Allow secret key cloning : 1
5: Allow secret key wrapping : 1
6: Allow secret key unwrapping : 1
7: Allow secret key masking : 0
10: Allow multipurpose keys : 1
11: Allow changing key attributes : 1
```

```

15: Ignore failed challenge responses : 1
16: Operate without RSA blinding : 1
17: Allow signing with non-local keys : 1
18: Allow raw RSA operations : 1
20: Max failed user logins allowed : 10
21: Allow high availability recovery : 1
22: Allow activation : 1
23: Allow auto-activation : 0
25: Minimum pin length (inverted: 255 - min) : 248
26: Maximum pin length : 255
28: Allow Key Management Functions : 1
29: Perform RSA signing without confirmation : 1
30: Allow Remote Authentication : 1
31: Allow private key unmasking : 1
32: Allow secret key unmasking : 1
33: Allow RSA PKCS mechanism : 1
34: Allow CBC-PAD (un)wrap keys of any size : 1
35: Allow private key SFF backup/restore : 1
36: Allow secret key SFF backup/restore : 1
37: Force Secure Trusted Channel : 0

```

Command Result : No Error

Observe that policy 22 is on; policy 23 is off, the result of creating the partition with partition policy template Sample02 as it exists at the moment.

4. Use command **partition policyTemplateList** to show the available partition policy templates:

```
partition policyTemplate list
```

Name	Description
Sample02	Another template
sample01	Sample partition policyTemplate

No partition policy template is currently loaded.

Command Result : No Error

5. Go back to the administrative slot if necessary, and use command **partition policyTemplateLoad** to load template Sample02 for modification:

```
lunacm:> partition policyTemplateLoad -name Sample02
```

Successfully loaded Sample02 partition policy template for editing.

Command Result : No Error

6. Use command **partition policyTemplateChange** to change policy 23 in the loaded (for editing) partition policy template:

```
lunacm:> partition policyTemplateChange -policy 23 -value on
```

Code	Description	Destructive
		Value Off-To-On On-To-Off

23	Allow auto-activation	On	No	No
----	-----------------------	----	----	----

Command Result : No Error

Observe that we can use the text string "On" or "Off" interchangeably with the numeric setting "1" or "0" to set a policy; both options are acceptable.

7. Use command **partition policyTemplateSave** to save the newly modified partition policy template with its modified policy value. Do not provide a name; the loaded policy already has one (in this case, "Sample02"):

```
lunacm:> partition policyTemplateSave
```

Saving the modified settings will overwrite the existing template "Sample02".

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

Sample02 successfully saved.

Command Result : No Error

8. Delete the previously-created demonstration partition, if necessary to make room.
Use command **partition create** with the **-policytemplate** option to create another new application partition, using partition policy template Sample02 previously created, and just now modified:

```
lunacm:> partition create -label parfortemplateyetagain -policyTemplate Sample02
```

Please attend to the PED.

Command Result : No Error

9. Use command **partition showpolicies** to show the policies of the new partition:

```
lunacm:> slot set slot 0
```

Current Slot Id: 0 (Luna User Slot 6.24.0 (PED) Signing With Cloning Mode)

Command Result : No Error

```
lunacm:> partition showpolicies
```

Partition Capabilities

```

0: Enable private key cloning : 1
1: Enable private key wrapping : 0
2: Enable private key unwrapping : 1
3: Enable private key masking : 0
4: Enable secret key cloning : 1
5: Enable secret key wrapping : 1
6: Enable secret key unwrapping : 1
7: Enable secret key masking : 0
10: Enable multipurpose keys : 1
11: Enable changing key attributes : 1
15: Allow failed challenge responses : 1
16: Enable operation without RSA blinding : 1
17: Enable signing with non-local keys : 1
18: Enable raw RSA operations : 1

```



```

20: Max failed user logins allowed : 10
21: Enable high availability recovery : 1
22: Enable activation : 1
23: Enable auto-activation : 1
25: Minimum pin length (inverted: 255 - min) : 248
26: Maximum pin length : 255
28: Enable Key Management Functions : 1
29: Enable RSA signing without confirmation : 1
30: Enable Remote Authentication : 1
31: Enable private key unmasking : 1
32: Enable secret key unmasking : 1
33: Enable RSA PKCS mechanism : 1
34: Enable CBC-PAD (un)wrap keys of any size : 1
35: Enable private key SFF backup/restore : 1
36: Enable secret key SFF backup/restore : 1
37: Enable Secure Trusted Channel : 1

```

Partition Policies

```

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

```

Command Result : No Error

Observe that both policy 22 and policy 23 are on (value = 1), as soon as the partition `parfortemplateyetagain` is created, using the recently-modified partition policy template "Sample02". For more information about those frequently-used policies, see ["About Activation and Auto-Activation" on page 268](#).

Note: The chosen partition affects the policies of a partition only when a partition is created.



In the examples on this page, partition `parfortemplateagain` was created when policy template `Sample02` was set to modify only partition policy 22. Therefore, partition `parfortemplateagain` does not have partition policy 23 set. The change to the policy template does not affect a partition that was already in existence. It has effect only for partitions that are created with that template after the template was modified.

Partition `parfortemplateyetagain` was created with the template after that modification, so it shows both policies changed.

You can change a policy manually, using **partition changepolicy** command.

Delete a partition policy template

If a partition policy template is no longer useful, use command **partition policyTemplate delete** to remove that template from the list.



Note: You must be in the administrative (HSM SO) slot in order to delete a partition policy template.

```
lunacm:> slot list
```

```
Slot Id -> 0
Tunnel Slot Id -> 2
Label ->
Serial Number -> 349297122742
Model -> K6 Base
Firmware Version -> 6.24.0
Configuration -> Luna User Partition With SO (PED) Signing With Cloning Mode
Slot Description -> User Token Slot

Slot Id -> 1
Tunnel Slot Id -> 2
Label -> mypcie6
Serial Number -> 150022
Model -> K6 Base
Firmware Version -> 6.24.0
Configuration -> Luna HSM Admin Partition (PED) Signing With Cloning Mode
Slot Description -> Admin Token Slot
HSM Configuration -> Luna HSM Admin Partition (PED)
HSM Status -> OK
```

```
Current Slot Id: 1
```

```
Command Result : No Error
```

```
lunacm:> slot set slot 1
```

```
Current Slot Id: 1 (Luna Admin Slot 6.24.0 (PED) Signing With Cloning Mode)
```

Command Result : No Error

lunacm:> partition policyTemplateList

Name	Description
Sample02	Another template
sample01	

No partition policy template is currently loaded.

Command Result : No Error

lunacm:> partition policyTemplateDelete -name sample01

Are you sure you wish to delete partition policy template: sample01

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

Successfully deleted partition policy template: sample01

Command Result : No Error

lunacm:> slot set slot 1

Current Slot Id: 1 (Luna Admin Slot 6.24.0 (PED) Signing With Cloning Mode)

Command Result : No Error

lunacm:> partition policyTemplateList

Name	Description
Sample02	Another template

No partition policy template is currently loaded.

Command Result : No Error

Separation of HSM Workspaces

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

- One for the Security Officer (SO)
- One for the Auditor, and
- One (or more) for applications and Clients.

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

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

Legacy Application Partitions

The application partition (or partitions, depending upon HSM type and configuration) is enabled (Activated) and managed by the partition User Owner in some regimes), and is then used by client applications to create and use cryptographic objects, and to perform cryptographic operations.

The ordinary partition User entity can be further sub-divided into Crypto Officer and Crypto-User in cryptographic security regimes that require this distinction. Legacy partitions are under the administrative control of the HSM SO, and do not have their own separate SO. The User or the Crypto Officer entity is created by the HSM SO.

PPSO Application Partitions

Either type (legacy or PPSO) can be created on an HSM with firmware 6.22.0 or newer and with the PPSO capability update installed. On HSMs that support multiple application partitions it is possible to create both types on the same HSM. A PPSO partition has its own SO. The Partition SO manages what happens inside its partition. The HSM SO creates the PPSO partition, and deletes it when necessary, but has no other oversight or control of the PPSO partition. This distinction is particularly important in cloud scenarios, but is a significant element in separation of roles for any use of an HSM.

Operation

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

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

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

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

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

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

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

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

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

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

Security Officer[0]

Crypto-Officer [1]

Crypto-User [2]:

Audit-User [3]:

Enter PIN :

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

- Crypto-Officer (which is the same as partition User (the black PED Key for PED-authenticated HSMs) if the Crypto-Officer/Crypto-User distinction is not in force) or
- Crypto User (which is the limited user when the Crypto-Officer/Crypto-User distinction has been invoked).

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

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

Key Management Commands

LUNA_CREATE_OBJECT:

LUNA_COPY_OBJECT:

LUNA_DESTROY_OBJECT:

LUNA_MODIFY_OBJECT:

LUNA_DESTROY_MULTIPLE_OBJECTS:

LUNA_GENERATE_KEY:

LUNA_GENERATE_KEY_W_VALUE:

LUNA_GENERATE_KEY_PAIR:

LUNA_WRAP_KEY:

LUNA_UNWRAP_KEY:

LUNA_UNWRAP_KEY_W_VALUE:

LUNA_DERIVE_KEY:

LUNA_DERIVE_KEY_W_VALUE:

LUNA_MODIFY_USAGE_COUNT:

Normal Usage Commands

LUNA_ENCRYPT_INIT:

LUNA_ENCRYPT:

LUNA_ENCRYPT_END:

LUNA_ENCRYPT_SINGLEPART:

LUNA_DECRYPT_INIT:

LUNA_DECRYPT:

LUNA_DECRYPT_END:

LUNA_DECRYPT_RAW_RSA:

LUNA_DECRYPT_SINGLEPART:

LUNA_DIGEST_INIT:
LUNA_DIGEST:
LUNA_DIGEST_KEY:
LUNA_DIGEST_END:
LUNA_SIGN_INIT:
LUNA_SIGN:
LUNA_SIGN_END:
LUNA_SIGN_SINGLEPART:
LUNA_VERIFY_INIT:
LUNA_VERIFY:
LUNA_VERIFY_END:
LUNA_VERIFY_SINGLEPART:
LUNA_GET_OBJECT_SIZE:
LUNA_SEED_RANDOM:

Unauthenticated Commands

LUNA_GET:
LUNA_GET_CONTAINER_LIST:
LUNA_GET_CONTAINER_NAME:
LUNA_LOGIN:
LUNA_OPEN_SESSION:
LUNA_PARTITION_SERNUM_GET:
LUNA_FIND_OBJECTS:
LUNA_GET_RANDOM:
LUNA_OPEN_ACCESS:
LUNA_GET_MECH_LIST:
LUNA_GET_MECH_INFO:
LUNA_SELF_TEST:
LUNA_GET_HSM_CAPABILITY_SET:
LUNA_GET_HSM_POLICY_SET:
LUNA_GET_CONTAINER_CAPABILITY_SET:
LUNA_GET_CONTAINER_POLICY_SET:
LUNA_GET_CONFIGURATION_ELEMENT_DESCRIPTION:
LUNA_RETRIEVE_LICENSE_LIST:
LUNA_QUERY_LICENSE:
LUNA_GET_CONTAINER_STATUS:

LUNA_GET_OUID:
LUNA_GET_CONTAINER_STORAGE_INFO:
LUNA_GET_ATTRIBUTE_VALUE:
LUNA_GET_ATTRIBUTE_SIZE:
LUNA_GET_HANDLE:
LUNA_INIT_TOKEN:
LUNA_PARTITION_INIT:
LUNA_CLOSE_ACCESS:
LUNA_DEACTIVATE:
LUNA_MTK_GET_STATE:
LUNA_MTK_RESPLIT:
LUNA_MTK_RESTORE:
LUNA_MTK_UNLOCK_CHALLENGE:
LUNA_MTK_UNLOCK_RESPONSE:
LUNA_MTK_ZEROIZE:
LUNA_CLEAN_ACCESS:
LUNA_PED_GET_SET_RAW_DATA:
LUNA_ZEROIZE:
LUNA_FACTORY_RESET:
LUNA_HA_LOGIN:
LUNA_CONFIGURE_SP:
LUNA_LOG_POLL_HOST:
LUNA_LOG_EXTERNAL:
LUNA_ROLE_STATE_GET:

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

LUNA_LOGOUT:
LUNA_CLOSE_ALL_SESSIONS:
LUNA_CLOSE_SESSION:
LUNA_GET_SESSION_INFO:

Configured and Registered Client Using an HSM Partition

Following the instructions in the previous sections, you have already:

- registered and assigned a Client to a SafeNet Network HSM Partition.

All that is required for a Client application to begin using a SafeNet Network HSM Partition (to which the Client has been assigned) is the standard handshake sequence:

- the **client establishes** a Network Trust Link connection with the SafeNet Network HSM (port 1792)
- the **client requests** a list of available Partitions (if not already known)
- **SafeNet Network HSM responds** with a list of only those Partitions to which the requesting Client has been assigned
- the **client chooses** a Partition from the available, assigned Partitions
- **SafeNet Network HSM demands** the password for the selected Partition
- the **Client** (which may also be called Crypto User if you are using the Crypto Officer / Crypto User authentication and access model) **provides** the appropriate password
- **SafeNet Network HSM grants** access, and the **Client application begins using** the Partition.

Your application should be capable of performing the above actions.

Simple Troubleshooting

If your Client application is having difficulty using SafeNet Network HSM for Client tasks, and if you have already verified the connection and the configuration (using multitoken and CMU utilities - see "[The Multitoken Utility](#)" or see "[About CMU Functions](#)"), then there may be a problem with the configuration of your Client application. Try the following suggestions before calling for support.

If your SafeNet Network HSM is a Password Authentication model, then you should look to your application setup for the source of the problem. It might require special configuration to use a Hardware Security Module (HSM). Or, if SafeNet Network HSM has replaced another HSM product (including a SafeNet product) you will need to modify the application to recognize the new device.



Note: Refer to the *SDK Reference Guide* and to the application integration documents provided by SafeNet Technical Support for information and advice on integrating many popular applications and services with SafeNet Network HSM.

However, if your SafeNet Network HSM is a PED Authenticated model, then be aware that having the Client application present the Partition Password is not sufficient to access the HSM Partition. The HSM Partition must also be in a special login state called activation (see), meaning that the Partition Owner (or Crypto Officer) must have logged in (with the correct black Partition Owner (or Crypto Officer) PED Key), and not logged out again before your application tried to connect. To ensure that the HSM Partition is always in the desired state, we recommend that you autoActivate (see "[About Activation and Auto-Activation](#) " below) the Partition, so that it can accept Client authentication and access at any time without human intervention at the SafeNet Network HSM appliance.

If you wish minute-by-minute control of a client's ability to access the HSM, without need for human presence at the appliance location, you could use the Remote PED feature (see "[About Remote PED](#)" on page 370).

About Activation and Auto-Activation

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

Activation

Activation is just a login with explicit caching of the login data, on the HSM.

- For legacy partitions, the cached authentication data is referred to as partition login data, handled by partition commands.
- For PPSO partitions, the cached authentication data is referred to as role login data, handled by role commands.

Login caching, or Activation, is convenient so that you can remove the black or gray PED key (perhaps to allow other uses of the PED, such as administrative logins by the HSM SO, or moving the PED to another HSM), while ensuring that access by Clients is not stopped, and that nobody is required to be present to press [ENTER] on the keypad on behalf of Clients.

To use Activation, you must first allow it by setting Partition Policy 22 (Allow Activation) to *on*, for each partition that you create. This is done by the HSM SO for legacy application partitions, and by the Partition SO for PPSO application partitions. If the Policy (22, Allow Activation) is *on*, then the partition Crypto Officer) can issue the `partition activate` command for legacy partitions. For PPSO partitions, once the policy is active it requires just **role login** to activate. The PED prompts for the black PED Key(s) and PED PIN if appropriate. Once you provide a black PED Key (Crypto Officer) or gray PED Key (Crypto User), the HSM appliance caches that authentication and the partition remains in a login state (Activated) until:

- you explicitly deactivate (with lunash command `partition deactivate`, or lunacm command **partition deactivate** or **role deactivate**, as appropriate)
- power is lost to the HSM.

You can remove the black PED Key (or gray PED Key) and keep it in your pocket or in safe storage. Activation remains on, and any registered Client with the Partition challenge password is able to connect and perform operations on the partition.

Activation is not a big advantage for Clients that connect and remain connected. It is an indispensable advantage in cases where Clients repeatedly connect to perform a task and then disconnect or close the cryptographic session following completion of each task.

To activate an application partition for use by registered Clients

1. Ensure that the partition policy "Allow activation" has been switched on.

For SafeNet PCI-E HSM or SafeNet USB HSM legacy application partition, type:

partition changepolicy -policy 22 -value 1

For SafeNet PCI-E HSM or SafeNet USB HSM or SafeNet Network HSM PPSO application partition, type:

partition changepolicy -slot <slot number> -policy 22 -value 1

2. To start activation of the desired partition, type:

partition activate -par <partitionname>

for legacy application partitions, or type:

role login -name <name of role to log in>

for PPSO application partitions.

Respond to the PED prompts.

AutoActivation

AutoActivation is supported for SafeNet Network HSM and for SafeNet PCI-E HSM, but not for SafeNet USB HSM.

AutoActivation extends the Activation feature, and allows automatic re-activation of the partition or the role, using the cached Crypto Officer or Crypto User authentication data, in the event of a restart or a short power outage (up to 2 hours). That is, the Activated state can recover to allow Clients to re-connect and continue using the application partition, without need for human intervention to insert the black PED Key (or gray PED Key) and press [ENTER] on the PED keypad.

AutoActivation, which you set by the `partition changePolicy` command, requires that Partition Policy 23 (Allow AutoActivation) be *on*, for the affected partition.

When you run the `partition activate` command for legacy partitions, or when you simply **role login** for PSO partitions, autoactivation is set as well (if you set policy 23 for that partition). You are directed to the PED, depending upon the current status of cached data.

If the authentication data requires refreshing, then the PED prompts you to insert the appropriate black or gray PED Key (that is, a PED Key that was imprinted with the partition authentication data for the particular partition [legacy] or role [PPSO]) and press [ENTER]. Once control returns to the command line, and the system announces success, you can remove the black PED Key and store it away. Clients can begin connecting and using the application partition.

We anticipate that most customers will set Partition Policy 23 Allow auto-activation (battery-backed caching of partition authentication) to *on* for their partitions, to ensure the convenience (uptime) of their clients.

Customers who prefer to not set auto-activation *on*, but who keep their SafeNet appliances located remotely from their administrative staff, might prefer to 'manually' resume partition activation by means of Remote PED. These options are entirely a matter of your preference and of your security policy.

To auto-activate an application partition for use by registered Clients

1. Ensure that Activation is switched on (see previous section).
2. Log in as the partition's administrator (HSM SO for legacy partition, Partition SO for PPSO partition).
3. Ensure that the partition policy "Allow auto-activation" has been switched on.

For SafeNet PCI-E HSM or SafeNet USB HSM legacy application partition, type:

```
partition changepolicy -policy 23 -value 1
```

For SafeNet PCI-E HSM or SafeNet USB HSM or SafeNet Network HSM PPSO application partition, type:

```
partition changepolicy -slot <slot number> -policy 23 -value 1
```

De-Activate a Partition

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

To deactivate an application partition

1. To deactivate, do one of the following.

For SafeNet PCI-E HSM or SafeNet USB HSM legacy application partition, type:

partition deactivate

For PPSO application partition on SafeNet PCI-E HSM or SafeNet USB HSM or SafeNet Network HSM, type:

role deactivate -name <name of role>

2. This turns off activation until the next time a login or activation is performed, at which time the authentication data is re-cached. Deactivation is temporary un-caching.

If you wish to turn off caching more permanently, so that it does not re-assert at the next login, change the Activation policy.

For SafeNet PCI-E HSM or SafeNet USB HSM legacy application partition, type:

partition changepolicy -policy 22 -value 0

For SafeNet PCI-E HSM or SafeNet USB HSM or SafeNet Network HSM PPSO application partition, type:

partition changepolicy -slot <slot number> -policy 22 -value 0

3. If Activation is turned off, then autoActivation is also turned off (cached authentication data is cleared) at the same time, but when the SafeNet HSM is Activated again, autoActivation resumes.
To turn off autoActivation, you must turn off the policy.

For SafeNet PCI-E HSM or SafeNet USB HSM legacy application partition, type:

partition changepolicy -policy 23 -value 0

For SafeNet PCI-E HSM or SafeNet USB HSM or SafeNet Network HSM PPSO application partition, type:

partition changepolicy -slot <slot number> -policy 23 -value 0

Removing Partitions

To remove an HSM Partition, you must be logged in (as admin) to the SafeNet appliance command shell (`lunash`) and you must be logged in to the on-board HSM as HSM Admin. Do the following:

```
partition delete -partition <HSM-Partition-name>
```

Replace `<HSM-Partition-name>` with the name of the Partition that you wish to delete (do not include the angle brackets "<>"). If you are not sure of the Partition name, use the `partition list` command.

When a Partition is **deleted**, the Partition is cleared from the HSM and any contents are deleted. This also implies that the Partition is revoked from any Clients that were registered to it.

By contrast, when a Partition is **revoked**, it still exists, but is no longer registered to (and available to) the Client from which it has been revoked. The Partition and its contents could still be used by other Clients, or could be re-assigned to the Client from which it was revoked.

Security of Your Partition Challenge

For SafeNet HSMs with Password Authentication, the partition password used for administrative access by the Partition Owner is also the partition challenge secret or password used by client applications.

For SafeNet HSMs with PED Authentication, the partition authentication used for administrative access by the Partition Owner is the secret on the black PED Key(s) for that partition. The partition challenge secret or password used by client applications is a separate random character string generated by SafeNet PED at the time the partition is created. This is one way in which we implement separation of roles in the SafeNet HSM security paradigm. The Partition Owner (holder of the black PED Key) can change the challenge secret dispensed by SafeNet PED for one that:

- is more "human-friendly", or
- is compliant with your organization's security policy (or is simply a different password/challenge in compliance with a mandated password-change interval)

How Secure Is the Challenge Secret or Password?

How secure do you want it to be?

When the question is asked, the concern is usually that a password harvesting attack of some sort might eventually crack the secret that protects the partition. Layers of protection are in place, to minimize or eliminate such a risk.

First, such an attack must be run from a SafeNet Client computer. That is not just any old computer. For interaction with HSM partitions on a SafeNet HSM network appliance, like SafeNet Network HSM, a SafeNet Client computer is one with SafeNet software installed, on which you have performed the exchange of certificates to create a Network Trusted Link (NTL). That exchange requires the knowledge and participation of the appliance administrator and the Partition Owner (who might, or might not, be the same person). That is, it is not possible to secretly turn a computer into a Client of a SafeNet HSM partition - an authorized person within your organization must participate.

Second, for SafeNet HSMs with Password Authentication, you set the partition password directly when you create the partition, so you can make it as secure as you wish (for an example of guidance on password strength, see <http://howsecureismypassword.net/> or <http://xkcd.com/936/>)

For SafeNet HSMs with PED Authentication, the partition password (challenge secret) is generated randomly, and displayed by the PED at partition creation and is therefore a very secure 16-character alphanumeric string that includes special characters.

Using the lunash:> command-line interface, you can change the partition password (or challenge secret) if you suspect it has been compromised, or if you are complying with a security policy that dictates regular password changes.

As long as you replace any password / challenge-secret with one that is equally secure, the possible vulnerability is extremely small.

Conversely, you can choose to replace a secure, random password/challenge-secret with one that is shorter or more memorable, but less secure - you assume the risks inherent in such a tradeoff.

Third, SafeNet HSM Partition Policy number 15 "Ignore failed challenge responses" can be set to "Off" (a value of zero). When that policy is off, the HSM stops ignoring bad challenge responses (that is, attempts to submit the partition secret) and begins treating them as failed login attempts. Each bad login attempt is counted.

Partition Policy number 20, "Max failed user logins allowed" determines how high that count can go before the partition is locked out.

Once a partition is locked by bad login attempts, it cannot be accessed until the HSM Security Officer (SO) unlocks it. So, for example, if you had set "Max failed user logins allowed" to 10, and if you had set "Ignore failed challenge responses" to Off, then an attacker on a client computer would have ten tries before the HSM stopped responding to his attempts. The attacker would then need to wait while the SO intervened to unlock the target partition. At that point, the attacker would have only ten more attempts until the cycle repeated. This defeats an automated harvesting attack that relies on millions of attempts occurring at computer-generated speeds. As well, after one or two such lockout cycles, the HSM SO would realize that an attack was under way and would rescind the NTL registration of the attacking computer. That computer would no longer exist as far as the HSM partition was concerned. The SO or your security organization would then investigate how the client computer had been compromised, and would correct the problem before allowing any new NTL registration from that source.

The above discussion illustrates that the degree and level of partition security is within your control. As the owner/administrator of the HSM, you get to determine any tradeoffs respecting security, convenience and other operational parameters. Via your security policies and procedures, you get to decide how much effort an attacker must expend; you control your response and your system's response to any potential attack.

Frequently Asked Questions

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

Why do I get an error when I attempt to set the partition policies for activation (22) and auto-activation (23) on my password authenticated SafeNet Network HSM?

Those policies apply to PED-authenticated SafeNet Network HSM, only.

For both PED-authenticated and Password authenticated HSMs, your client authenticates to a partition with a challenge password.

For PED-authenticated HSMs, the application partition must be in a state where it is able to accept that challenge password. That is, the extra layer of authentication - the partition Crypto Officer's black PED Key or the Crypto User's gray PED Key - must have been presented first before the partition can be receptive to the challenge/password.

Password-authenticated HSMs have only the single layer of authentication - the challenge/password is all that is needed. The password is both the client authentication and the partition administrator (Crypto Officer / Crypto User) authentication.

For PED-authenticated HSMs, activation and auto-activation enable caching of the first layer of authentication to provide a level of operational convenience similar to that of the Password-authenticated HSMs.

So, what is the difference in security, once Activation and Auto-activation are started?

From the convenience point of view, none. But, whereas the Password-authenticated partition is "open for business" to anybody with that partition's password, as soon as the partition is created, a PED-authenticated partition is not. One implication is that all partitions of a multi-partition password-authenticated HSM are available whenever any of them are available, which is essentially whenever the HSM is powered on.

The owner of a PED-authenticated HSM partition can disable client access to just one partition by deactivating (de-caching) just that one partition's PED Key authentication, so that the challenge/password is not accepted. Any other partitions on that HSM that are not deactivated (i.e., still have their black PED Key or gray PED Key authentication cached) are still able to accept challenge/password from their clients.

You are not required to cache the PED Key data in order to use a partition. You could, if you preferred, simply leave the PED Key for that partition inserted in a connected SafeNet PED, and press keypad keys on the PED whenever first-level authentication for partition access was required. Since this would defeat much of the reason for having a powerful networked HSM server, generally nobody does this with SafeNet Network HSM in a production environment. (For the kind of application where that scenario might be appropriate, we recommend a host-installed SafeNet PCI-E HSM or a USB-connected SafeNet USB HSM.) As well, if you have created both a Crypto Officer and a Crypto User for your partition, you would need to switch out the black PED Key or the gray PED Key, whenever the other entity needed to PED-authenticate, while the PED Key authentications are not cached.

You also have the option of partially engaging the PED Key caching feature by enabling Activation without enabling Auto-activation. In that case, you present your PED Key to activate the partition - which allows it to accept its partition challenge/password from clients - and the cached black PED Key or gray PED Key authentication data is retained while the HSM has power (or until you explicitly de-cache). But the cached authentication does not survive a power outage or an intentional power cycle (because you chose to Activate, but not to AutoActivate as well). Thus, by applying different policy settings, you could have some partitions on your PED-authenticated HSM able to return to client availability immediately following a power-cycle/outage (no human intervention needed), while others would wait for your intervention, with a black PED Key (Crypto Officer) or a gray PED Key (Crypto User), before becoming client-available.

Finally, Activation and Auto-activation are partition-level policy settings, not role-level. Therefore, if the policy is on, it is on for all roles. If the policy is off, it is off for all roles. You cannot individually cache authentication data from a gray PED Key, but not from a black PED Key (or the opposite) within a single partition.

HSM Status Values

Each HSM administrative slot shown in a slot listing includes an HSM status¹. Here are the possible values and what they mean, and what is required to recover from each one.

Indicated Status of HSM	Meaning	Recovery
OK	The HSM is in a good state, working properly.	n/a
Zeroized	The HSM is in zeroized state. All objects and roles are unusable.	HSM initialization is required before the HSM can be used again. "Hard init" - HSM SO and domain are gone, no authentication required. (see Note1)
Decommissioned	The HSM has been decommissioned (zeroized and factory reset).	HSM initialization is required before the HSM can be used again. "Hard init" - HSM SO and domain are gone, no authentication required. (see Note1)
Transport Mode	The HSM is in Secure Transport Mode.	STM must be disabled, by providing the correct purple PED Key, before the HSM can be used. (see Note2)
Transport Mode, zeroized	The HSM is in Secure Transport Mode, and is also zeroized.	STM must be disabled, by providing the correct purple PED Key (see Note2), and then HSM ("hard") initialization is required before the HSM can be used.
Transport Mode, Decommissioned	The HSM is in Secure Transport Mode, and has been decommissioned (zeroized and factory reset).	STM must be disabled, by providing the correct purple PED Key (see Note2), and then HSM ("hard") initialization is required before the HSM can be used.
Hardware Tamper	The HSM has been tampered. (MTK is destroyed and must be rebuilt from recovery vectors.)	Reboot the host or restart the

¹The state or condition of a device, as reported in the user interface.

Indicated Status of HSM	Meaning	Recovery
		HSM (vreset for SafeNet PCI-E HSM, or ureset for SafeNet USB HSM). The event is logged. If one of the recovery vectors is external (on a purple PED Key) then you are prompted to provide it before the HSM can recover from tamper. Resume using the HSM. (see Note2)
Hardware Tamper, Zeroized	The HSM has been tampered. (MTK is destroyed and must be rebuilt from recovery vectors.) The HSM is also in zeroized state. All objects and roles are unusable.	Reboot the host or restart the HSM (vreset for SafeNet PCI-E HSM, or ureset for SafeNet USB HSM). The event is logged. If one of the recovery vectors is external (on a purple PED Key) then you are prompted to provide it before the HSM can recover from tamper. (see Note2) HSM initialization is required before the HSM can be used again. "Hard init" - HSM SO and domain are gone, no authentication required. (see Note1)
HSM Tamper, Decommissioned	The HSM has been tampered. (MTK is destroyed and must be rebuilt from recovery vectors.) The HSM has also been decommissioned (zeroized and factory reset).	Reboot the host or restart the HSM (vreset for SafeNet PCI-E HSM, or ureset for SafeNet USB HSM). The event is logged. If one of the recovery vectors is external (on a purple PED Key) then you are prompted to provide it before the HSM can recover from tamper. (see Note2) HSM initialization is required before the HSM can be used again. "Hard init" - HSM SO and domain are gone, no authentication required. (see Note1)

NOTE1: A condition, not reported above, preserves the HSM SO and the associated Domain, while SO objects and all application partitions and contents are destroyed. HSM SO login is required to perform the "soft init".

Indicated Status of HSM	Meaning	Recovery
-------------------------	---------	----------

See ["Initialization Overview for PED-authenticated HSMs" on page 240](#) for more information.

NOTE2: If the HSM is placed in Secure Transport Mode, or if the HSM experiences a Hardware Tamper event while a recovery vector is external to the HSM, and you are unable to provide the requested purple PED Key (with that external recovery vector), then the HSM is unrecoverable. Contact SafeNet to obtain an RMA and ship the HSM back for re-manufacture.

(Applies to PED-authenticated HSMs only.)

If your HSM is Password-authenticated, or if your PED-authenticated HSM has both recovery vectors internal (no purple PED Key was created), then if a tamper event destroys the MTK, the HSM recreates the MTK after being restarted, and no further intervention is required.

The above scenarios assume that a tamper event is transient, and the cause is corrected. If the HSM remains in tamper, or immediately returns to tamper, then contact SafeNet Technical Support.

For comparison and detailed explanation of "hard init" vs "soft init", see ["Initialization Overview for Password-Authenticated HSMs" on page 238](#) and ["Initialization Overview for PED-authenticated HSMs" on page 240](#).

For a comparison of various destruction or denial actions on the HSM, see ["Comparison of Destruction/Denial Actions" on page 400](#).

Key Migration

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

- "Key Migration Procedures" below
- "Migrating Key Material from Older (2U) to New (1U) Appliances" below
- "Frequently Asked Questions" below

Key Migration Procedures

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

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

Migrating Key Material from Older (2U) to New (1U) Appliances

Contact SafeNet Technical Support at support@safenet-inc.com or www.safenet-inc.com/Support

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.

We want to migrate from a Microsoft Certificate Authority to a Linux CA while keeping the same private key. Does the SafeNet HSM offer any barriers to doing this?

This is not recommended. It is not an issue of the HSM. Rather it is an issue of the software that you use to run your CA. When you generate a key in the HSM, it is stored internally in a partition. The key can be used by any application that has appropriate access and can successfully authenticate to the partition. That application could be the Microsoft CA, a Linux CA, or both, or other.

Most applications expect to generate and control their own key material. If your Linux CA allows you to point at an HSM and say "use that key", then the SafeNet HSM does not prevent it from doing so. However, as an example, when Microsoft CA creates the root key, it embeds a representation of the machine name in the key, to enforce that only that machine can have access to it. In the Microsoft world it is possible to get around this obstacle by using "clusterkey", but it is not clear how the Linux CA would react, as we have not tested such configurations.

Generally, the "best practice" that we recommend for switching from one PKI platform to another, or from one HSM vendor to another, is to implement the new PKI with a new root and issuing CA key, while leaving the original PKI in place. All new certificates are issued from the new PKI, and the original PKI (no longer used to issue certificates) is allowed to phase out over time, as each certificate that was issued from it expires.

PED Authentication

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

- "About the SafeNet PED" below
- "Using the PED" on the next page
- "About PED Keys" on page 288
- "What is a PED PIN?" on page 296
- "How to Use a SafeNet PED" on page 305
- "Interaction between the HSM and the PED" on page 306
- "Lost PED Keys or PED PINs, or passwords" on page 315

About the SafeNet PED

SafeNet 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 SafeNet PED 1.x - contact Gemalto Technical Support.

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.

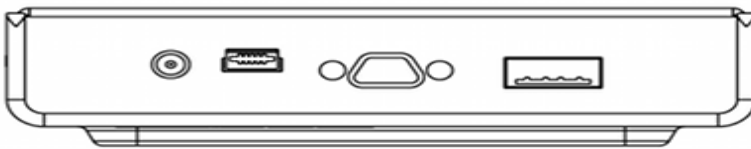
As well, you require a set of at least three PED Keys. For PED 2.0 and later, the PED Keys are in the form of hardware identification tokens, SafeNet iKey model 1000 (RoHS-compliant) or possibly other SafeNet iKey models, to be introduced at a later date. For most applications, you would want an additional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

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-adaptor 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 (local-only) PED 2 and the Remote Capable PED 2 is a sticker on the upper right-front panel, either local

PED local

or remote

PED remote

Using the PED

A SafeNet PED is required to authenticate to an HSM that requires 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 one exception is the SafeNet Backup HSM, which configures itself, at backup time, as either Password-authenticated or PED-authenticated, depending on the type of primary HSM it is backing up).

Figure 1: PED (2.x) front view

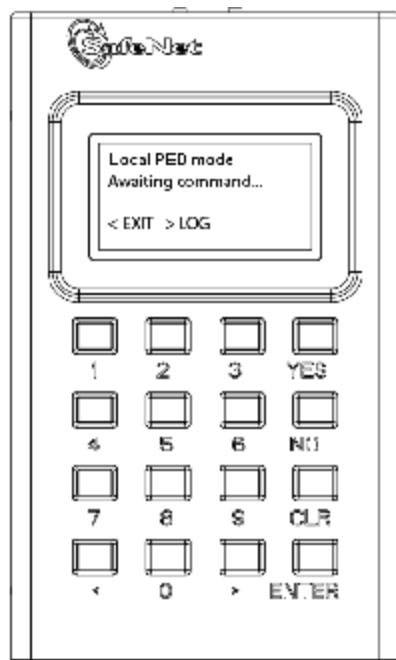
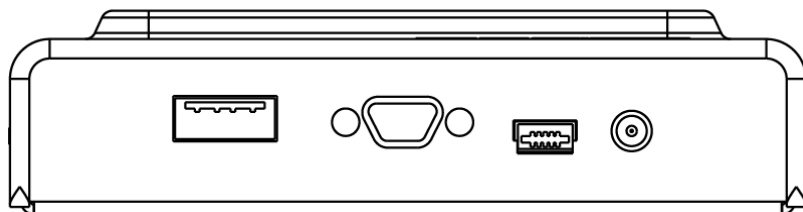


Figure 2: PED top view



Interaction with Other Operations

HSM firmware version 6.24.0 introduces a change in how ongoing PED operations interact with cryptographic operations requested simultaneously.

Behavior before HSM firmware version 6.24.0

PED operations interrupt other operations occurring at the same time on the HSM. The HSM waits for a PED operation to complete before processing requests for other operations. This can cause delays in production.

Behavior after HSM firmware version 6.24.0

PED operations no longer interrupt other operations occurring at the same time on the HSM in most cases. The most beneficial effect is that PED operations acting on a partition no longer block operations occurring on other partitions on the same HSM. For example, you can now create new partitions or backups while running cryptographic operations on a separate partition. In this way, you can perform maintenance and configuration on your HSM without interrupting important client applications. PED operations might still block cryptographic operations occurring on the same partition, especially high volumes of write object requests.

Versions

PEDs are generally unit-interchangeable (with limitations within the version range, PED 2.x, see table), and more specifically interchangeable within the same PED-firmware version. That is, if a SafeNet PED with a given firmware supports your current operation with your current HSM version, then any SafeNet PED with the same, or newer, firmware can replace it.



Note: Exception - If you are using the Remote PED feature, only another PED with Remote capability can support that operation, regardless of firmware version.

Newer PED firmware versions are compatible with HSM versions shown in their row in the table, and backward compatible with any earlier HSM that requires a version 2.x PED.

PED firmware version	Local PED operation and Remote PED capable	PED-mediated MofN per secret (with HSM f/w 6.x) also SRK (purple PED Key) and Secure Transport Mode	Field update-able	Audit User (white PED Key)	Small Form-factor Backup	PED version is feature-compatible with SafeNet HSM firmware version(s)
2.2.0	Yes	No	No	No	No	SafeNet HSM 4, f/w 4.x
2.4.0-3	Yes	Yes	To 2.5.0	No	No	SafeNet 5.0, f/w 6.0.8 SafeNet 5.1.x, f/w 6.2.1
2.5.0-3	Yes	Yes	To 2.6.0	Yes	No	SafeNet 5.2, f/w 6.10.2 SafeNet 5.3.1 f/w 6.20.0
2.6.0-6	Yes	Yes	Yes	Yes	Yes	SafeNet 5.4, f/w 6.21.0 SafeNet 6.0, f/w 6.22.0

PED firmware 2.2.0 is mentioned in the table above because many customers who first bought SafeNet HSM 5.0 were already in possession of older PEDs since they already had earlier SafeNet HSMs (f/w 4.x). SafeNet HSM 5.0 needed PED f/w 2.4.0-3 to access all functions.

PED firmware 2.5.0-3 or newer is suitable for all local and remote authentication and is required for some PED-mediated features added since SafeNet HSM 5.0.

PED firmware 2.6.0-x, available as a field upgrade or on newly-purchased PEDs, supports SafeNet Small Form-Factor Backup - a completely separate function mediated by SafeNet PED, and using different USB tokens - and also supports all previous PED 2.x authentication functions.

Authentication

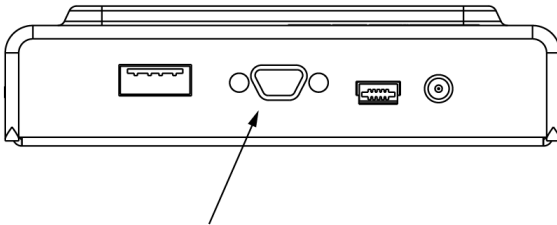
In this current discussion, we ignore SFF Backup, and focus on the HSM authentication function of SafeNet PED. The authentication information for your HSM roles is contained on the PED Key, and SafeNet PED is the device that provides the interface so that authentication data can pass between PED Key and HSM (see ["About PED Keys" on page 288](#)).



The keypad on the PED is used to acknowledge prompts (on the PED screen) and to optionally input a "something you know" additional secret, called a PED PIN (see ["What is a PED PIN?" on page 296](#)) to augment the "something you have" secret contained in the PED Key.

Local and Remote

A locally-connected PED is powered by its connection to the HSM.



That connection - directly between the PED and the HSM card inside the host - bypasses your computer bus and the computer bus of the HSM host (if separate). 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 see ["What is a PED PIN?" on page 296](#)), 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 and host computers often tucked away in server farms, which are frequently 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 SafeNet 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. A Remote PED does not have the direct connection to an HSM to provide power; it uses a USB connection for data exchange, which might not provide sufficient stable power for operation. Therefore a PED used in Remote mode also needs a dedicated power connection via the provided power block.

For both local and remote PED use, the only way for another person to discover a PED PIN password while you are inputting it is if you allow that person to observe 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 shell (lunash on SafeNet Network HSM) or Lunacm (for any SafeNet HSM) 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 activation, 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.

For Remote PED connections, the MDSM connector is not used, and power and USB connections are used instead.

What Do I Do?

As soon as it receives power from a connection to a powered HSM, or from the supplied power block if you are using Remote PED, the PED performs its start-up and self-test routines and then goes to its normal operating mode, displaying the prompt "Awaiting command...". The PED is ready for use in Local mode, by default.

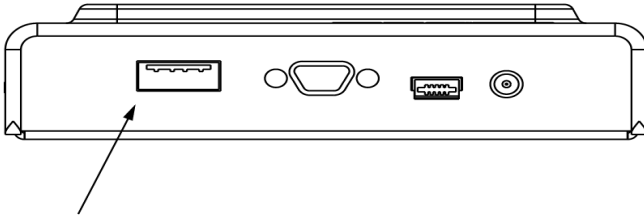
There are three things that you can do with the PED at this point:

- Wait for a prompt, which results when a program has caused the HSM to request authentication
- Change to the Remote Mode (which expects encrypted commands from a computer USB connection, where you would be running Pedserver, rather than from a direct PED-HSM connection)
- 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 numeric passwords (PED PINs) 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, multitoken, or other SafeNet utilities), or operations triggered by your applications.



The operations "Initializing a PED-Authenticated HSM" on page 1 and "Prepare to Create a Partition (PED Authenticated)" on page 1 are described elsewhere in this documentation.

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	Blue, red and black PED Keys and PED. Optionally a purple PED Key, if you or someone invoked Secure Transport Mode, and an orange PED Key if an RPK was already created, and you are performing these actions remotely.	You perform the HSM initialization, possibly create Partition Groups with other HSMs, set up a redundant, load-sharing HA group 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 to their assigned HA group partitions	Clients need their own authentication that is set up before client applications are launched to use a partition or an HA group virtual partition; no PED Key or PED required, but you need the Crypto Officer and/or Crypto User challenge secrets.	None. You would normally have activated the individual HSM or (in case of HA) the HA-group members (in other sections of this table), 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 Remote PED, 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 Local PED mode.
2. Press "4" to enter Admin mode.
3. In Admin mode, options are 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.) Press "1".
4. To perform an operation on a particular PED Key, insert that PED Key into the PED Key connector on top of the PED.
5. 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 do not need login.
6. 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 PED Key to another (blank or re-used) PED Key, press "7" on the PED keypad.
7. When prompted, insert a blank target PED Key, or a non-blank whose data is no longer needed, and press ENTER.
8. 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.
9. 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.
10. 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: Secure Recovery

The PED will not perform a standalone copy operation (that is, without an HSM) of a purple PED Key. This is a security feature. You can copy a purple PED Key (just like any other PED Key for any other HSM role or function) during an imprinting operation controlled by a SafeNet HSM. Because purple PED Keys are specific to a single HSM, no other HSM can share a purple key or make a copy. The PED refusal to make standalone copies of purple keys is just an additional barrier to anyone wanting to attack an HSM that has been placed in Secure Transport Mode.

EXCEPTION: Remote PED

The Remote PED 2 functions as described earlier, 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. The remote functionality is described separately at "[About Remote PED](#)" on page 370.

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. See "[About FIPS Validation](#)" on page 459 for more information.

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

SafeNet HSM
Security
Officer

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 keypad) can be added. SO PED Keys can be duplicated¹ for backup, and can be **shared among HSMs** by imprinting subsequent HSMs with an SO PIN already on a PED Key.

Partition User or Crypto Officer

SafeNet HSM
Crypto Officer

Application Partition User key or Crypto Officer key. For HSM SO-controlled application partitions (with no SO local to the partition) this PED Key is required to log in as HSM Partition Owner or Crypto Officer. Needed for Partition maintenance, creation and destruction of key objects, etc. Creates the optional Crypto User.



Note: Creation of a challenge secret is forced for an application partition owned by the HSM SO (a.k.a. "Administrator").

For application partitions that have their own SO, this PED Key is required to log in as Crypto Officer, and is needed for Partition maintenance, creation and destruction of key objects, etc. The Crypto Officer creates the Crypto User.

The challenge secret generated in conjunction with the Crypto Officer can grant client applications access to create, delete, and manipulate partition objects.



Note: Creation of a challenge secret is not forced for an application partition with its own SO.


¹a 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 for tracking of the "paper trail" of possession, to satisfy your security auditors.

A PED PIN (an additional, optional password typed on the PED keypad) can be added. Black Crypto Officer PED Keys can be duplicated¹ for backup, and can be shared among application Partitions using the "Group PED Key" option.

Crypto User



The Crypto User has restricted read-only administrative access to application partition objects. The challenge secret generated in conjunction with the Crypto User can grant client applications restricted, sign-verify access to partition objects.

 **Note:** Creation of a challenge secret is forced for an application partition owned by the HSM SO. Creation of a challenge secret is not forced for an application partition with its own SO.

A PED PIN (an additional, optional password typed on the PED keypad) can be added. Gray User PED Keys can be duplicated² for backup, and can be shared among application Partitions using the "Group PED Key" option.

Domain



Key Cloning Vector (KCV) or Domain ID key. This PED Key carries the domain³ 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 operations (see [What is a Domain PED Key?](#) later in this section, for a more detailed explanation). A PED PIN (an additional, optional password typed on the PED keypad) can be added at the time the PED Key is created/imprinted. Red PED Keys can be duplicated⁴ for backup or multiple copies of the key.

¹a 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 for tracking of the "paper trail" of possession, to satisfy your security auditors.

²a 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 for tracking of the "paper trail" of possession, to satisfy your security auditors.

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.

⁴a 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 for tracking of the "paper trail" of possession, to satisfy your security auditors.

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 access the HSM host 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 SafeNet PED connected to a properly configured computer to substitute for a PED connected directly to the SafeNet appliance/HSM, when that local connection is not convenient.

The RPV is created/imprinted by a SafeNet 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 SafeNet 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 SafeNet 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 SafeNet appliance/HSM with PED Keys and PED. Orange PED Keys can be duplicated¹ 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 Tamper 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 Tamper Key, allowing the HSM to be used.

A PED PIN (an additional, optional password typed on the PED keypad) can be added. Purple PED Keys can be duplicated² for backup or multiple copies of the key, but only during the creation/imprinting of the key - SafeNet PED cannot duplicate purple keys via the standalone PED admin menu. Purple PED Keys are unique to each HSM, and cannot be shared.

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

¹a 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 for tracking of the "paper trail" of possession, to satisfy your security auditors.

²a 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 for tracking of the "paper trail" of possession, to satisfy your security auditors.

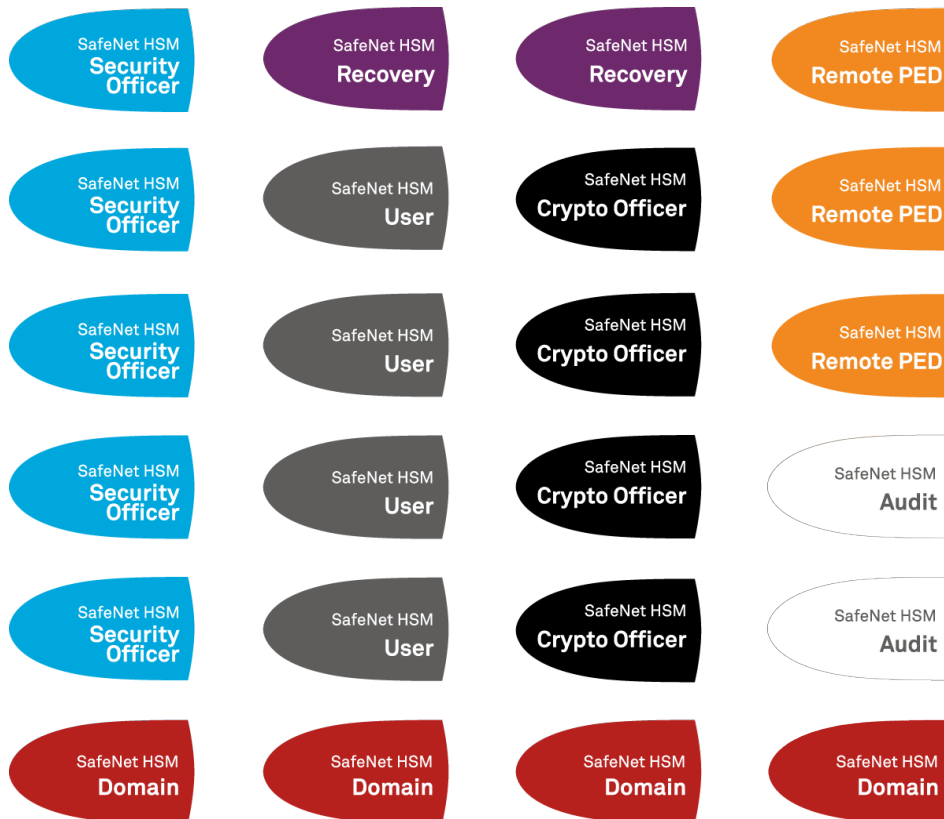
as other logging parameters, such as rollover period, etc. The purpose of the separate Audit role is to satisfy certain security requirements, while ensuring that no one else - including the HSM SO - can modify the logs or hide any actions on the HSM. The Audit role is optional until initialized.

For SafeNet USB HSM and SafeNet PCI-E HSM see the audit commands in lunacm:>.

A PED PIN (an additional, optional password typed on the PED keypad) can be added. White Audit PED Keys can be duplicated¹ for backup, and can be shared among HSMs using the "Group PED Key" option.

What is a Set of PED Keys?

A nominal set of **blank** 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 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.

The set of stickers/labels does not indicate a requirement. The quantity of each color on a sheet is merely an average distribution according to customer practices that we have seen.

You are not required to have a PED Key of each color, above. The ones that are mandatory for use of your PED-authenticated HSM are the blue SO key, the red domain key, and the black Crypto Officer key.

¹a 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 for tracking of the "paper trail" of possession, to satisfy your security auditors.

- Operation of the HSM does not require that you create a Crypto User - it is optional, in case you need a limited-capability client.
- Operation of the HSM does not require that you create an Audit user - if your situation does not require audit logging, that is, if standard, unverified HSM logging is sufficient, then the Audit role is superfluous to your needs. On the other hand, if your implementation is likely to be audited against security standards, then creating, assigning, and using the HSM's Audit role would be recommended.
- Operation of the HSM does not require that you imprint a Remote PED Key, as long as you have local access for PED interactions, when needed. If you expect to administer the HSM remotely, then a Remote PED Key will be required.
- Operation of the HSM does not require that you imprint a Secure Recovery PED Key. If no external Secure Recovery Vector is generated, then the HSM can recover from tamper events with only a login, and the HSM cannot be placed in Secure Transport Mode. If you wish to invoke (and later recover from) Secure Transport Mode, or if you require that the HSM halt in case of a tamper event, and require intervention (presentation of a purple PED Key) to acknowledge and recover from such tamper, then enabling SRK would be necessary.

Imprinting

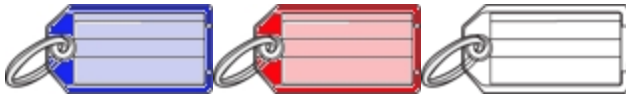
- A PED Key can contain only one HSM authentication secret at a time.
- PED Keys are completely interchangeable before they are imprinted by your action - the PED checks for an existing authentication secret, and tells you if the currently presented key is blank.
- PED Keys are imprinted by SafeNet PED during HSM initialization and Partition creation and other HSM actions that create HSM roles or invoke certain HSM functions.
- PED Keys can be re-imprinted with new HSM authentication secrets. Imprinting a new secret overwrites (destroys) any HSM authentication secret that was already present on a PED Key when it was presented for new imprinting. The PED always warns you if the presented key contains an authentication secret. The PED has no way to know if an authentication secret that it finds on a key is useful and valid for some role on the current HSM (or on some other HSM), or if a contained authentication secret is outdated and no longer valid, and therefore safe to overwrite - so the PED simply tells you what it has found and lets you make the decision.
- PED Keys are completely interchangeable for re-imprinting (except see Note, below) - that is, you can turn any PED Key into a different PED Key during an initialization/imprinting operation; the PED warns you that the key you have presented already contains an authentication secret (and tells you what kind), but you can choose to overwrite if you think the currently imprinted secret is no longer useful.

Note: The exception is the purple SRK PED Key. If you attempt an operation that creates and imprints a new SRK value, the PED will accept any blank key or any non-purple, previously-imprinted PED Key to accept the new external recovery vector for the currently-connected HSM. That means, it will imprint a new SRV onto any blank key, and it will also imprint a new SRV onto any key that currently contains authentication for SO, CO, CU, Domain, Auditor, RPK, if you tell it to overwrite.



But, it will not overwrite a purple PED Key that contains the currently valid SRV for the current HSM. This is a safety feature.

Keep in mind that the PED has no way to determine if a discovered SRV secret on a key is currently valid for some other HSM. It can check only with the currently-connected HSM.



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 in SafeNet PED's own dialogs.

- Security Officer (SO) key - blue
- domain key - red
- Crypto Officer key (or partition Owner/ key, old scheme) - black
- Crypto User key - gray
- 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 (a backup token is initialized/re-initialized whenever a backup is performed onto it). 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 (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). 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 MofN security feature.

The numbers of PED Keys that you will need for your situation are discussed in much greater detail at ["How Many PED Keys Do I Need?" on page 333](#).

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.

If you invoke the optional MofN security feature (see ["Using MofN" on page 343](#)), you could have multiple sets of several PED Keys (containing the secret splits for SO, or for the Partition Owner, or for Domain, etc.) that might require unique visible identification. Possibly one person might be the designated holder of MofN 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 MofN 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 MofN feature, your key tags might be labeled something like:



So this would be Security Officer (SO) key-share number 4, of a 5-key MofN 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:



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 SafeNet PED" on page 305](#).

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.

Compare Password and PED Authentication

	Password-authenticated HSM	PED-authenticated HSM
Ability to restrict access to	<ul style="list-style-type: none"> knowledge of Partition Password is sufficient 	<ul style="list-style-type: none"> ownership of the black PED Key is mandatory

	Password-authenticated HSM	PED-authenticated HSM
cryptographic keys	<ul style="list-style-type: none"> for backup/restore, knowledge of partition domain password is sufficient 	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> not available 	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> linked to password knowledge, only 	<ul style="list-style-type: none"> linked to partition password knowledge, linked to black PED Key(s) ownership
Role-based Access Control (RBAC) - ability to confer the least privileges necessary to perform a role	roles limited to: <ul style="list-style-type: none"> Auditor HSM Admin (SO) Partition Owner 	available roles: <ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> not available 	<ul style="list-style-type: none"> Remote PED and orange (Remote PED Vector) PED Key deliver highly secure remote management of HSM, including remote backup

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.

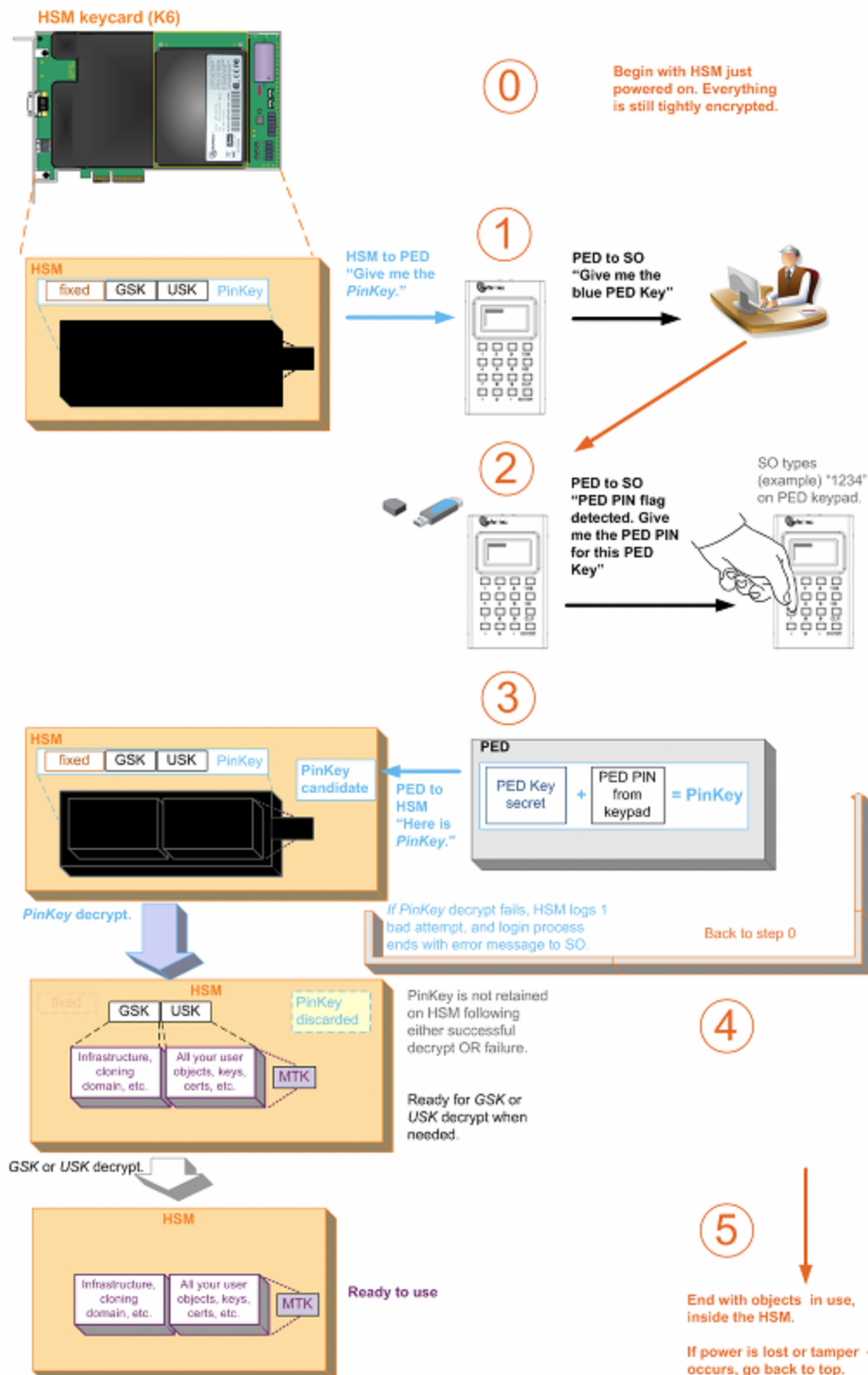
In Password authenticated HSMs, the PinKey is the text password that you type at a keyboard.

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.

HSM – PED interaction, one PED PIN



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 SafeNet PED prompts you to provide a PED PIN. (see below)

How to invoke/require a PED PIN with an HSM

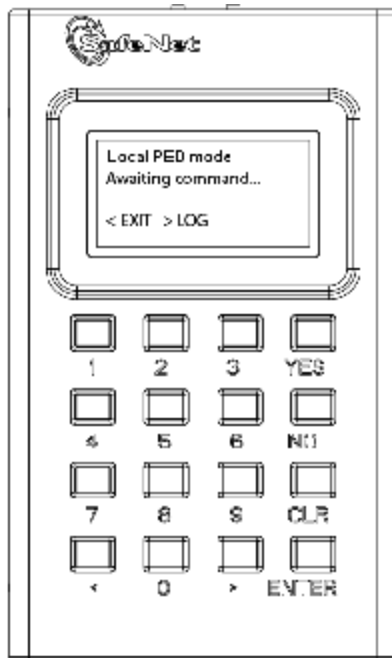
At the SafeNet PED prompt:

Enter a new PED PIN

If you want a PED PIN:

- enter 4 to 48 digits via the SafeNet [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

- must have physical possession of the PED KEY,
- must have physical access to the HSM and PED, and
- 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
- SafeNet 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 SafeNet 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.

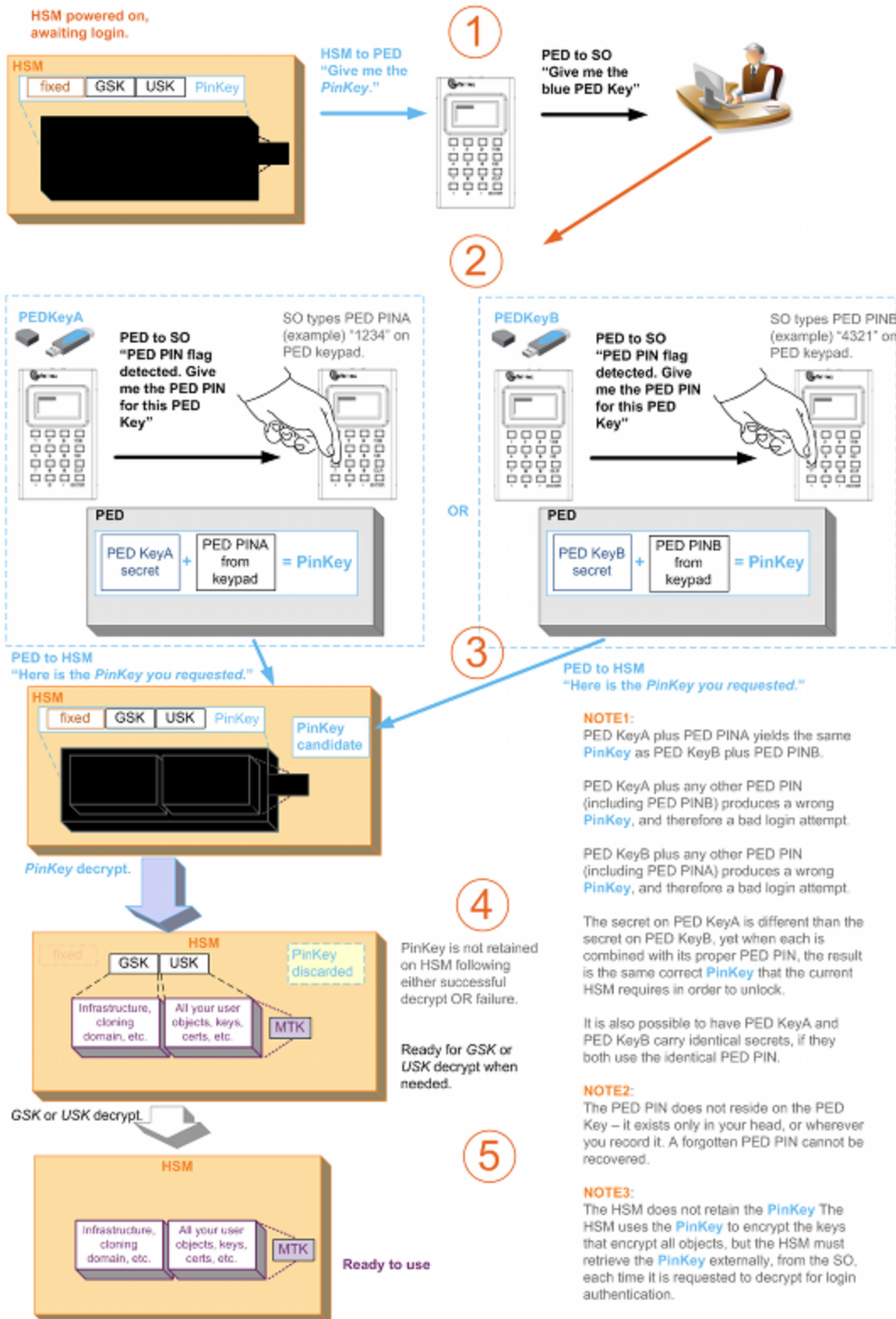
Here are possible combinations if you have two HSMs H1 and H2, and any of several initialization-time choices regarding PED PIN. What is important to unlock the HSM is the secret that is imprinted on the HSM, so in the following table we will call that secret H1SO or H2SO. We will call the secret contained 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	H1SO secret identical to	Either PED Key plus the correct PED PIN for just that PED Key; both are correct for either HSM. Either PED Key with the PED	Yes, but the PED PINs are not.

Configuration	SO Authent Secret on HSM	What You Need to Unlock HSM	PED Keys Interchangeable?
generate the same PinKey secret as PED Key2 plus PED PIN2 - H1SO = H2SO	H2SO	PIN for the OTHER PED Key is a bad login attempt.	

Here is a drawing of HSM PED authentication with two PED PINs.

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



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 secret¹.

Note: 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.

¹[which is now invalid for the current HSM]

Does that apply to the other PED Key colors?

Not all.

- It does apply to the black PED Key - use the lunash 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 (lunash command `hsm 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 lunash command `hsm 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:

"Shared or Group PED Keys" on page 326

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 [you answered "NO" to the "...reuse..." question], 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).

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).

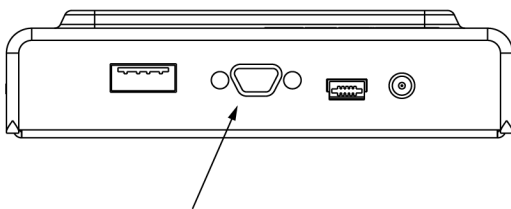
In the case of a red key, that would mean you would need to attempt a cloning or backup/restore operation before storing your record of the PED PIN.

How to Use a SafeNet PED

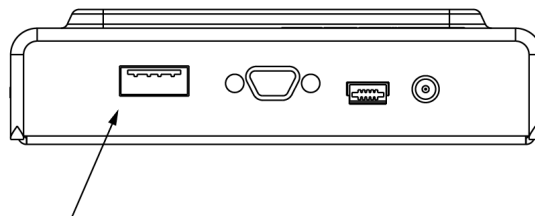
The PED, when used locally, derives its power from its connection to a SafeNet HSM.

To use the SafeNet PED:

1. Connect the SafeNet PED to the PED connector on the SafeNet HSM, using the supplied cable.



2. SafeNet 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 SafeNet PED operation, the SafeNet 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 SafeNet 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.



Note: The SafeNet PED display returns to "Awaiting command.." when the current sequence of PED operations is finished.
 "Awaiting command.." on the SafeNet PED means that control has been transferred back to the HSM.

SafeNet PED Keypad Functions

Key	Function
[CLR] or [Clear]	<ul style="list-style-type: none"> - 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.
[<]	<ul style="list-style-type: none"> - 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	<ul style="list-style-type: none"> - 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..."

SafeNet PED Interaction

Go to "Interaction between the HSM and the PED" below to read about using the SafeNet PED with your HSM.

Remote PED

Got to "About Remote PED" on page 370 to read about using a SafeNet PED remotely from your SafeNet HSM, via PedServer and PedClient.

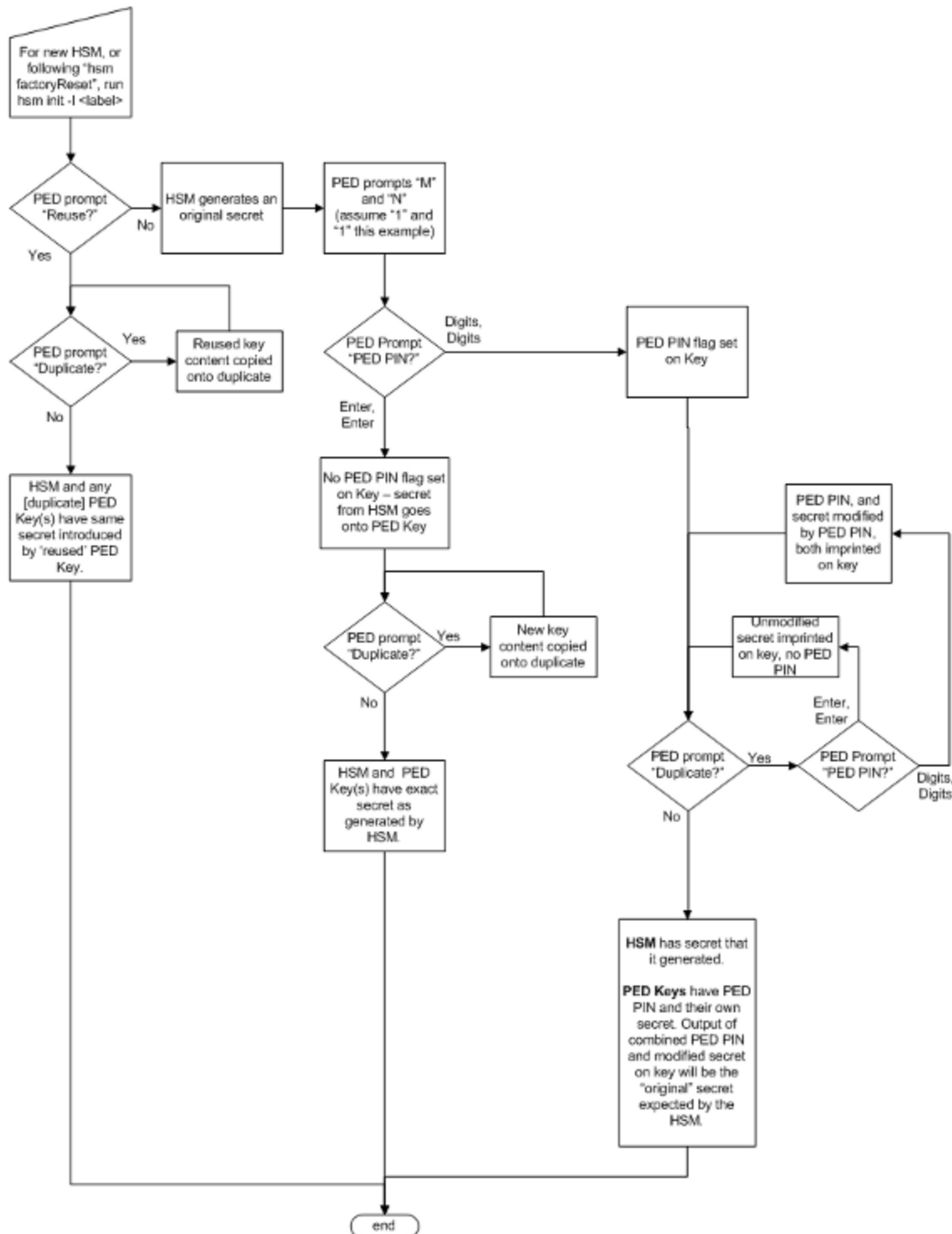
Interaction between the HSM and the PED

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

After the first-ever SafeNet 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)

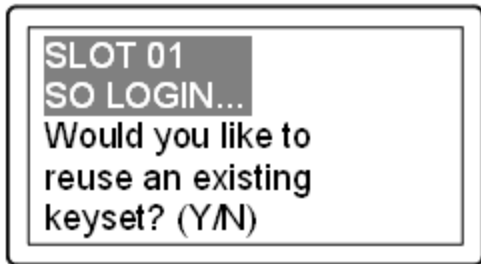
interact with SafeNet PED and PED Keys, particularly during initialization - a time when important decisions are made. Other pages describe the PED and PED Keys. This is more about flow.

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.



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

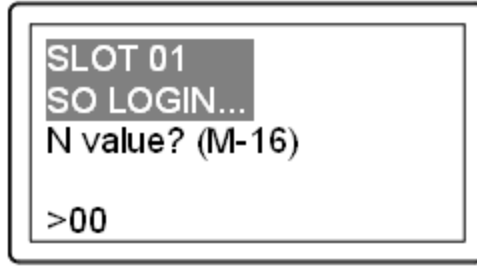
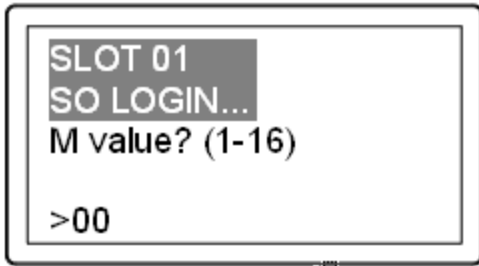
The HSM slots that form an HA group interact with their client(s) via a virtual HSM slot, such that any of the real HSM slots behind the HA group is interchangeable and can be swapped in and out as needed. But members of an HA group do not need to be members of a PED Key group. In an HA group, any or 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,



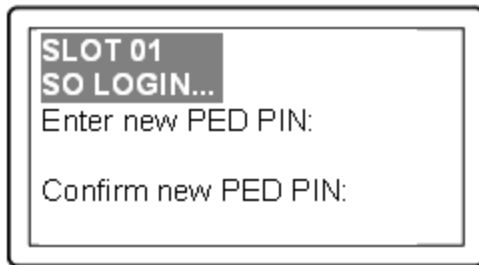
so that you could set

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".

If you wish to see an explanation of how MofN works on the HSM, go to "[HSM Authentication Model with MofN and No PED PINs](#)".

PED PIN?

The PED provides the opportunity to add an additional layer of authentication security to the handling of the current secret.

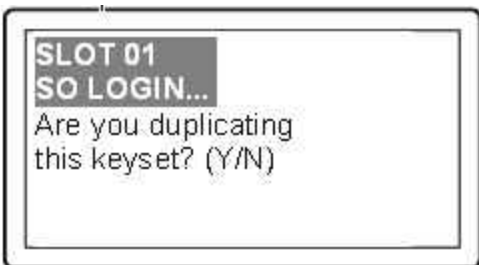


A PED PIN is a numeric secret typed on the PED keypad. If you just press enter, no PED PIN is created, and therefore no PED-PIN flag is set on the current PED Key. If you do type in some digits on the PED's keypad, then that sequence becomes a PED PIN, a numeric password that must be typed whenever you 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.

If you wish to see an explanation of how PED PINs work on the HSM go to "[HSM Authentication Model with a PED PIN](#)" or "[HSM Authentication Model with Multiple PED PINs](#)".

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 SafeNet Network HSM 6.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) MofN, split-secret, multi-person access control model.

Historical

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 (SafeNet PCI-E HSM 5, SafeNet Network HSM 5, SafeNet USB HSM), 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 MofN) 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. Reusing, or not, is chosen on a per-role basis, so that **some** PED Key secrets on an HSM could be shared with other HSMs, while others are not.
- It is crucially important to always control your PED Keys. Know where they are, know how many there 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 [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?"]. 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 head¹ 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) 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 Crypto Officer (black key) secret cannot be "reused" as an HSM SO (blue key) secret. Nor can a Domain (red), or RPK (orange), 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

SafeNet PED has the ability to make copies of PED Keys, without the intervention of an HSM. All the PED needs is power.

¹[or wherever you write it down]

Duplicating PED Keys / Copying PED Keys

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:

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.

Compare Duplication via PED Admin 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 or 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 SafeNet 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 SafeNet 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.

PED Key Management

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

- "PED Key Management Overview" below
- "PED Keys and Operational Roles" on page 322
- "Actions That Require a PED Key" on page 324
- "Shared or Group PED Keys" on page 326
- "Domain PED Keys" on page 328
- "Duplicating PED Keys" on page 331
- "How Many PED Keys Do I Need?" on page 333
- "Using MofN" on page 343
- "Complexity When Managing PED Keys" on page 347
- "General Advice on PED Key Handling" on page 347
- "Updating PED Keys – Example" on page 349
- "Updating PED Key for a Backup Token" on page 353
- "Frequently Asked Questions" on page 272

PED Key Management Overview

This section applies to SafeNet 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. For any role on the HSM, options like "group"/reuse, MofN, or the use of PED PINs (second factor of two-factor authentication) are imposed, or not, at the time the role is created.

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).

"Possible" Does Not Mean "Necessary"

When you initialize an HSM or create a Partition, SafeNet 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 Administrator 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

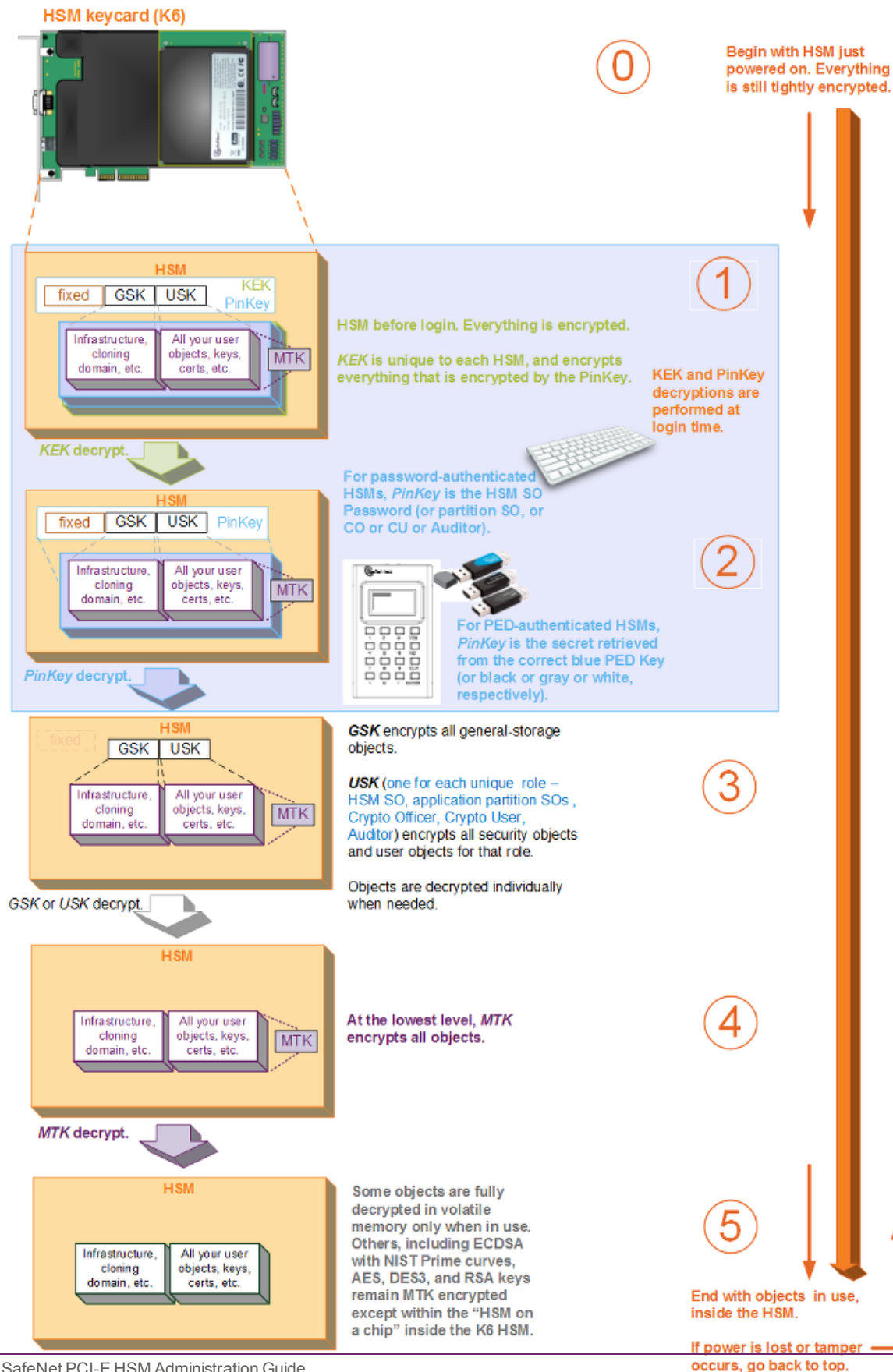
access that HSM. This can be done in one of two ways:

- 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 SafeNet 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 SafeNet HSMs
(you do this when you answer "YES" to the "Reuse an existing keyset" question from SafeNet PED)

HSM Layered Encryption - the General Case



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). 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 SafeNet 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 "Are you duplicating 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.




Note: 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. See "General Advice on PED Key Handling" on page 347 for more information.










Note: The person or persons charged with ownership of the HSM, are responsible to safeguard the authentication secrets, ensuring that no unrecorded duplicates are made. Similarly, for application partitions with their own SO, the SO of each partition is responsible for securing the authentication secrets and copies.

PED Keys and Operational Roles

Below are some suggested holders of PED Keys by role.

Lifecycle	PED Key [Note 1]	Operational Role	Function	Custodian
<i>PED keys enforce division of operational roles and prevent unilateral action by key holders</i>				
HSM Admin		Security Officer	Manages provisioning activities and global security policies for the HSM : - HSM initialization, - partition provisioning, - global policy for the HSM and the partitions within it.	CSO CIO
		Domain Cloning	Cryptographically	Domain Administrator
SafeNet PCI-E HSM Administration Guide				

Lifecycle	PED Key [Note 1]	Operational Role	Function	Custodian
		Token Backup	defines the set of HSMs or partitions that can participate in cloning for the purposes of backup and high-availability.	WAN Administrator
		Secure Recovery	Restores an HSM after a Secure Transport or tamper event	CSO
		Remote PED	Establish a Remote PED connection	System Administrator
Application Partition Admin		Security Officer	Manages provisioning activities and global security policies for the partition : - partition initialization, - role setting, - policy setting.	
Daily Operation		Crypto Officer	This is the full user role associated with a partition. This role can perform both cryptographic services and key management functions on keys within the partition.	System Administrator
		Crypto User	This is a restricted user role on a partition. This role can perform cryptographic services using keys already existing	System Administrator

Lifecycle	PED Key [Note 1]	Operational Role	Function	Custodian
			within the partition, only. (See Note 2, below.)	
Ongoing Auditing		Audit User	An independent role responsible for audit log management. This role has no access to other HSM services.	Auditor
<p>[Note 1: This table implies a single PED Key for each HSM role or functional secret. For any role or PED Key secret, you can elect to invoke the MofN split-knowledge shared secret option, to spread the responsibility for that role or function over multiple persons. That is, you can require that a predetermined number of responsible persons, greater than one, must be present to unlock/access the particular HSM role or function. Choose MofN for a role or function when it is important that no single person have unsupervised access. See "About MofN" on page 1 and "Using MofN" on page 343.</p>				
<p>[Note 2: Functionally, the Crypto User (gray¹) PED Key is just another "black PED Key". The PED does not distinguish gray from black. The gray label is provided only for your convenience, so that CO and CU PED Keys are easy to visually identify and manage.</p> <p>It is useful to have two separate PED Keys (one for each of CO and CU) for separation of those administrative roles, in which case two different color labels are helpful for physical identification and handling. But if that administrative separation is not important in your setting, you can use just a single black key that authenticates to both roles, and still have two separate challenge secrets to give to applications:</p> <ul style="list-style-type: none"> - one for applications that need read-write crypto access to your partition, and - one for applications that are allowed only read-use access.] 				

Actions That Require a PED Key

It can, occasionally, be less than obvious why a certain action requires that you authenticate to the HSM or to the Partition, while another action does not.

Such questions have been carefully considered, from a crypto-security perspective, and we believe that we have consistently made the correct determination in all cases.

An example might be:

Question

If I activate an existing partition - make a service available to customers - I must have the black PED Key for that partition

But if I want to deactivate the same partition - withdraw/deny the service - I do not need the black key).

Is making a service available considered more dangerous than taking it away?

¹An alternate spelling of "grey". If you see either "gray" or "grey" throughout these documents, they refer to the same concept.

Answer

The rationale behind this behavior is that when you activate a partition you are making crypto services available to applications (that have the correct challenge password, of course). From a crypto module perspective, making crypto services available is a big thing and requires proper authentication. Removing that availability might be an operational issue but it is not a crypto security issue and, therefore, did not require the Black key.

As well, if an attacker wishing to deny service is given physical or command access to your HSM, he or she can do a lot more damage than simply issuing a `partition deactivate`. In other words, if you have let them get that deeply inside your security perimeter, then you have far worse problems than a "partition deactivate".

If you ever discover a situation where our implementation seems inconsistent with that philosophy, please let us know by contacting support@safenet-inc.com. We will either fix the problem or explain why it is not considered a problem.

Question

If I have a service running, can I force my application's administrator to provide the partition's password each time the service is restarted and/or each time the application server is restarted?

Answer

It depends on the setup, and it depends on what you mean by password. But, in limited circumstances, yes, although you probably would not want to do that.

It doesn't really matter whether your application accesses the HSM directly, or whether a service or some other provider is between application and HSM.

If an application directly accesses the HSM partition, then when it first does so, the application must initialize the library, and open a session on the HSM. Then the application provides the partition authentication when the application needs to perform actions on partition contents. For either Password-authenticated or PED-authenticated HSMs, the partition password (or partition challenge) secret must be available to the application so that it can provide that secret when access to partition objects is needed. The application provides the partition secret to say that it (the application) has the right to perform partition-object actions. This usually means that the secret is stored somewhere on the host's file system or registry (most likely encrypted) for retrieval when needed.

This means that the application is already providing the partition password (or partition challenge secret) string whenever it is demanded by the HSM. So, in that sense, the answer to your question is already "yes".

But if you meant something stronger than the password-presenting action that the application must already perform when accessing partition objects, you probably need to consider PED authenticated HSMs.

For PED-authenticated HSMs, the PED Key data for that partition must be provided to the HSM before the partition secret string is provided.

In almost all cases, for PED-authenticated HSMs, customers would Activate the partition when first setting up, so the PED Key data for that partition would be cached. That is, the customer would be making an authenticated administrative declaration that the partition was "open for business", and an application with the partition challenge secret could then access partition objects at any time.

Then, the application could open and close sessions at will, and whenever it needed to manipulate partition objects, the application would provide the partition (challenge) secret. Once the Partition PED Key data is available, the action of accessing and using partition objects is identical for PED-authenticated or Password-authenticated HSMs.

If the partition is autoActivated, then the black PED Key data is cached in the HSM, just as for Activation, except it is now protected against power failure for as much as two hours.

So, for the direct application-to-HSM scenario, if you want to force the application owner to perform an authentication beyond what the application already performs with each access to partition objects, then you would need:

- a. to use a PED-authenticated HSM, and

- b. ensure that the PED Key data for that partition was NOT cached - therefore, no Activation or autoActivation (Partition Policies 22 and 23 would be set to "off").

The application still has its access to the partition challenge secret, but the partition is not "open for business". A PED Key must be provided (possibly a PED PIN as well, if you set one).

However, the drawback is that EVERY access of partition objects now requires PED Key authentication, in addition to the partition challenge secret. The PED would remain connected to the HSM, the key for that partition would remain inserted, and somebody would have her/his finger poised to press the PED's [Enter] key every time the application needed to manipulate a partition object.

The above is the situation for direct access to the HSM by an application; it is only for very specialized situations where the partition is rarely accessed, and extremely close control is required.

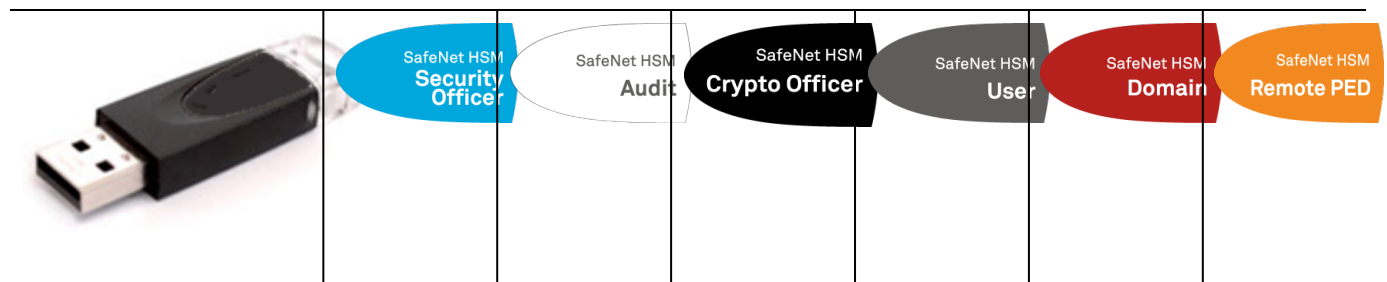
If you insert a service between your application(s) and the HSM, then the application no longer needs to know anything about HSM and its authentication. Instead, the service must handle all that, translating between the application and the HSM. The conditions described earlier now apply to the service.

Similarly, if you insert a provider (a translation layer like our CSP or KSP or JSP) between your application and the HSM, or between your service and the HSM, then the provider takes care of safeguarding and using the partition password (or partition challenge) secret as needed.

In either case, the need for PED Key presentation and PED button pressing is determined by the state of Partition Policies 22 and 23).

See also ["Commands that Require SafeNet PED Interaction"](#).

Shared or Group PED Keys



With the common administrative group option (answer "YES" to the SafeNet 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 SafeNet 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 SafeNet HSM with PED [Trusted Path] Authentication), SafeNet 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 SafeNet 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 SafeNet 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

SafeNet HSM
Recovery

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 an SRV split of its Master Tamper Key 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 secret.

(marked with)



and imprinted with a domain

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 SafeNet Backup HSM, and by which restoring is possible from a Backup HSM or token to a SafeNet 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.

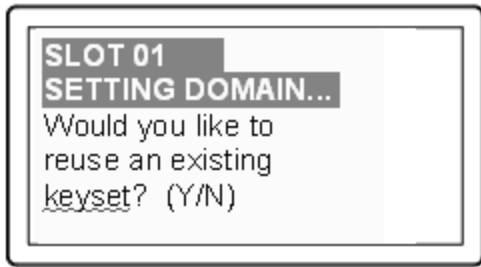
At initialization time, the key-cloning vector is created on the HSM and imprinted onto a red PED Key, or if a desired cloning domain already exists, then the existing key-cloning vector from a red PED Key is read from that PED Key and imprinted on the HSM (or Backup token) as the HSM (or token) is initialized. 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 SafeNet 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, SafeNet PED first asks:



If you answer [No]:

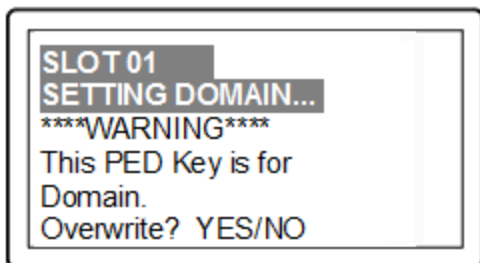
- You are telling SafeNet PED that it should retrieve a new domain (Key Cloning Vector) from the HSM and prepare to overwrite that new domain secret onto a blank key that you are about to insert, or overwrite the existing random domain vector on a red PED Key that you are about to insert.
- This was 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 device), then that HSM (as well as the Backup device) 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]:

- SafeNet PED prepares to preserve 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 device
- 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, SafeNet 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, SafeNet PED needs to know whether to retain it. SafeNet PED asks:



If you answer Yes:

- SafeNet 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 device), then that HSM (as well as the Backup device) 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 No:

- SafeNet 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, SafeNet 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 SafeNet Network HSM 6.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.

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 SafeNet Network HSM) and G5 (the HSM for PKI with SafeNet Network HSM, and the core of the SafeNet Backup HSM) - legacy HSMs - used an older, smaller domain secret, which is incompatible with current HSMs.

Cloning of objects between SafeNet 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.

Summary

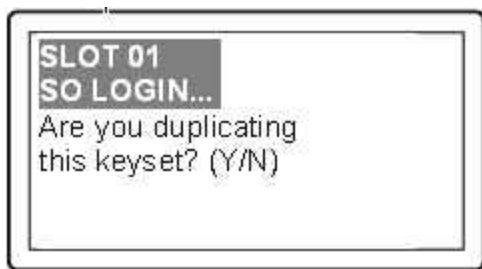
If you can account for all the HSMs to which you have presented your red Domain PED Key (meaning that you have maintained strict control of that red PED Key), then you know with certainty that nobody else could possibly have a copy of the sensitive keys that were created on your HSMs or partitions, or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

Duplicating 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:



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 SafeNet PED's Admin menu.

Considerations for Duplicate PED Keys

The duplicate PED Key option permits you to issue (or store) more than one PED Key (duplicates) for any of:

- HSM Admin

- Auditor
- Remote PED vector
- Secure Recovery vector
- Owner PED Key (legacy),
- Partition SO (PPSO),
- Crypto Officer or Crypto User per HSM Partition.

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 some-or-all PED Keys for an 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 stringent. 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 of the relevant PED Key will be needed by each person who must ever have access to any given HSM Partition, and to each person with HSM Admin/SO 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 SafeNet 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.

Implications of Duplicate PED Keys

By implication, your security and operational procedures must ensure that no person takes advantage of that facility to make unauthorized or un-tracked copies of any key.

The SafeNet PED (and the associated HSM) do not know how many copies you have made, so you are given the option every time you initialize an HSM or create a role or secret, 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 on-board admin menu of the SafeNet 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. If somebody loses an imprinted PED Key, the person who finds it has potential access to your HSM, in that role. However, if a misplaced (or stolen) imprinted PED Key also has a PED PIN associated with it, then it would be much more difficult for the finder to make use of the found/stolen PED Key.

What a duplicate PED Key is Not

Duplicate PED Keys are not the same as MofN-split PED Keys. Whatever secret is on a PED Key that you duplicate is the secret that is contained on the duplicate(s). If you selected "M value" and "N value" to be 1 (one) when creating the first PED Key, then there is no splitting of the secret, therefore any duplicate of that key is also a complete, self-contained copy of that secret, and either the original or the duplicate is fully sufficient to authenticate. If you choose to split a secret when creating it, by selecting "M value" and "N value" greater than 1 (one), then a duplicate of that secret must create duplicates of all the splits.

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.

Basic amount for operation

The basic amount is described in the topic/page ["About PED Keys" on page 288](#). If you elect not to make use of some of the roles and functions, then you have fewer to manage than the seven (legacy) or eight (with Per-Partition SO) that are described there.

The next question is: How many HSMs do you have? If you have just one SafeNet HSM, then there is no need to consider other HSMs when you determine your security/role/authentication policy. If you have more than one SafeNet HSM, you need to make decisions about how they are to be administered and who will do it. These issues are discussed and illustrated in the following sections.

Regarding HSM authentication, the factors to consider are:

- How many HSMs are to be administered by one identity (are you making a group PED Key for several HSMs, or a unique PED Key per HSM)?
- Does your security policy allow a single person to control access to HSM functions, or should that access be divided, requiring multiple authorized persons to oversee authentication to an HSM role (["About MofN" on page 1](#), or ["Many PED Keys for one HSM - MofN" on page 336](#) below)?
- Will these decisions apply to all HSMs within your sphere?
- Will these decisions apply to all, or to just some roles on each HSM?
- For each role or authentication secret related to an HSM, how many operational and on-site- and off-site-backup copies must be created and maintained and tracked?

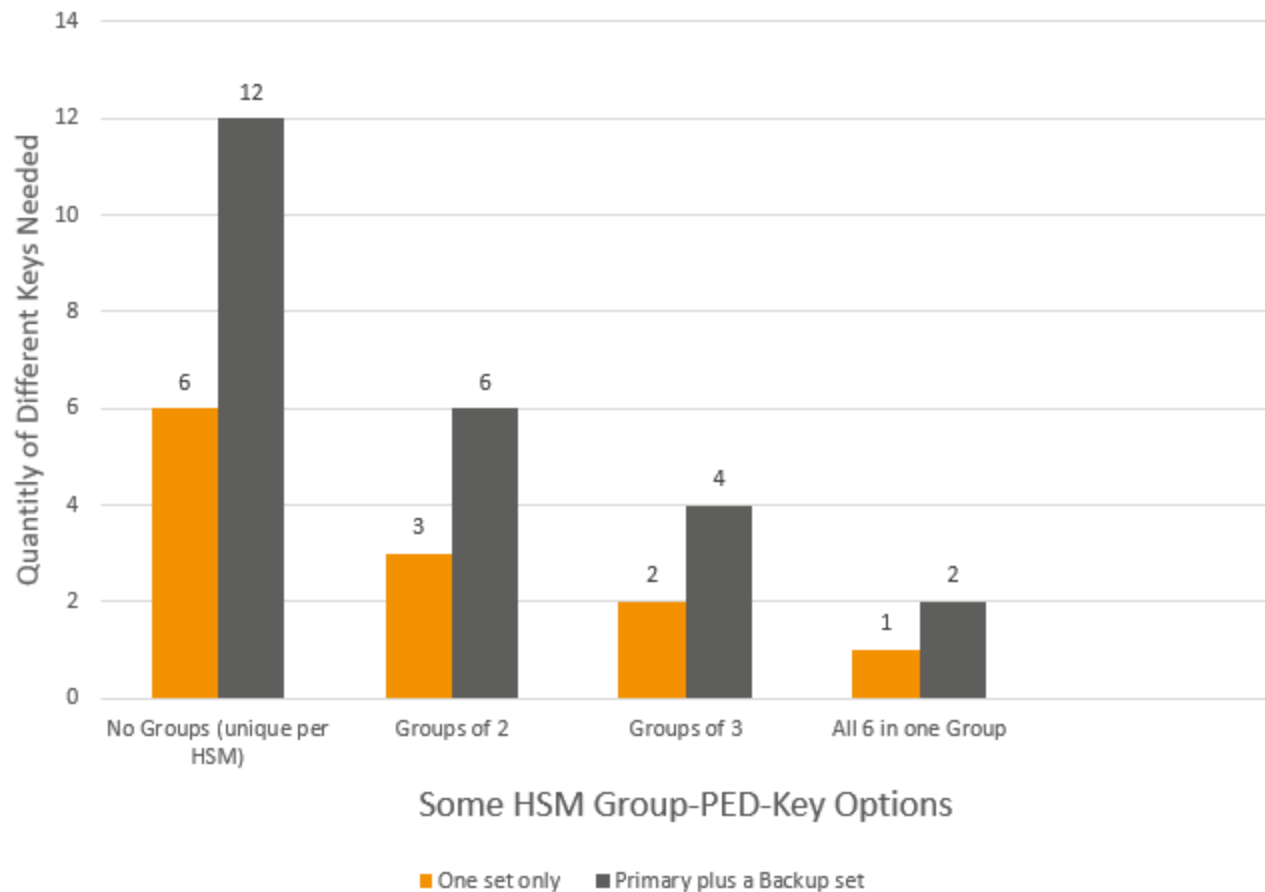
One PED Key for many HSMs - grouping or reuse

This section discusses the number of on-site full sets required, for different choices regarding PED Key reuse/grouping, and then considers the implications of having backup authentication keys. For this discussion, assume that you have more than one HSM to administer - for this example, we arbitrarily assume six, but the concepts could apply to any number.

The following example considers grouping of PED Keys, which is the outcome of your response to the SafeNet PED prompt "Do you wish to reuse a keyset? Y/N". You can imprint a new, unique secret onto both the HSM and the current PED Key, by responding "No" to the reuse question. This is the "no group" or the "group of 1" scenario.

Alternatively, you can accept a currently-valid PED Key and imprint that secret onto the HSM, such that the current PED Key can unlock its role or function on the current HSM as well as on one or more HSMs that already have the same secret (for that role or function). This is the "grouping" scenario, where one PED Key is good for multiple HSMs, which eases the administrative burden if your security policy so permits.

PED Keys Needed - 6 HSMs with Grouping



The above chart depicts some options for the "...reusing a keyset..." question from SafeNet PED when you are imprinting an HSM authentication secret. The example shown is an organization responsible for six (6) SafeNet HSMs.

The first column of each pair in the chart, in orange, depicts how many keys you would need for a given secret :

- On the left, "No Groups", if you set a unique secret for each of the six HSMs (responding "No" every time SafeNet PED asked if you were reusing a keyset; therefore 6 different keys, one for each HSM). Six keys are needed for that one secret.
- If you chose to imprint authentication secrets onto your six HSMs in three groups of two HSMs (responding "No" when asked if you were reusing a keyset, for the first HSM of each group, then responding "Yes" when imprinting the secret for the second HSM of each group). Three keys are needed for that secret.
- If you chose to imprint authentication secrets onto your six HSMs in two groups of three HSMs (responding "No" when asked if you were reusing a keyset, for the first HSM of each group, then responding "Yes" when imprinting the secret for the second HSM, and again for the third HSM of each group). Two keys are needed for that secret.
- If you chose to imprint authentication secrets onto your six HSMs in one group of six HSMs (responding "No" when asked if you were reusing a keyset, for the first HSM of the group, then responding "Yes" when imprinting the secret for the second HSM, and again for the third HSM, and again for the fourth HSM, and again for the fifth HSM, and again for the sixth HSM of the group). One key is needed for that secret.



Note: Yes, you could also do a group of five and a group of one, or a group of two and a group of four... any combination that works in your operational environment and satisfies your security requirements.



Note: So far, we are discussing just one authentication secret, out of seven or eight for any HSM. We would need to repeat this discussion for each of the other secrets, per HSM that you manage.

However, those quantities, enumerated above, imply that you have just one set of PED Keys for the HSMs that you need to access. This is unwise; if you lose or damage the only key of a given color (for a given role or feature), you lose access to that aspect of that HSM or that group of HSMs. With that in mind, the second column of each pair in the chart, in gray, represents how many physical PED Keys you would need in each situation for reliable access to your six HSMs, if you have one working (primary) set, as well as an additional, backup set of keys.

Many organizations would go further, and insist on having three complete sets, one primary or working set, one backup set to be stored in on-site secure lockup, and a third set to be stored in off-site secure lockup. This kind of requirement might be specified or implied in a Business Continuity/Disaster Recovery plan, or in a comprehensive security policy.

Again, with each HSM and with each authentication secret or secured function on that HSM, at the time the secret is created or reset, the PED gives you the opportunity:

- to "Reuse an existing keyset" - make the current HSM secret part of an existing group that is unlocked by an already-imprinted PED Key (or an already-imprinted MofN keyset), or
- to **not** "Reuse an existing keyset", and thereby use a fresh, unique secret generated by the current HSM that is not shared by any other HSM, meaning that this is the opportunity to start a new group, independent of any existing group of HSMs.

The same issues arise with respect to all PED-mediated secrets (HSM SO, partition SO, cloning domain, Crypto Officer and Crypto User, Remote PED, Auditor). The exception is the purple Secure Recovery PED Key, which is unique to its HSM and cannot be grouped.

The other special case is the red Cloning Domain PED Key, which **must** be shared where you wish to be able to clone contents among HSMs or among application partitions.

Secrets are independent for grouping

The different secrets are independent. You could group SO authentication keys but keep partition Crypto Officer authentication keys unique. You could maintain unique SO and Crypto Officer authentication keys for all your HSMs and their partitions, while at the same time having a single, grouped Cloning Domain for all... or for some.

You could decide that some combination of grouping and uniqueness was suitable for all your HSMs and their partitions and roles, but that you wanted a single Auditor identity for all HSMs in your organization.

You could implement any combination of unique and grouped authentication secrets that suited your organization's working methods and security policies.



CAUTION: Always have at least one complete set of backup PED Keys in addition to the working or operational set.

The above sections have discussed deploying HSM authentication secrets on a one-for-one (unique PED Key per HSM) or a one-for-many (grouped PED Key for multiple HSMs) basis, or any of several possible combinations.



Note: HSM grouping, or PED Key reuse, helps to reduce the number of PED Keys that must be tracked and managed in your organization. If you have multiple HSMs, and if there is no operational or security reason to maintain unique authentication for each, then grouping several HSMs to be accessed by one PED Key is a useful option.

The next sections discuss a different option, regarding PED Keys, that can have considerable effect on the number of PED Keys that you must manage.

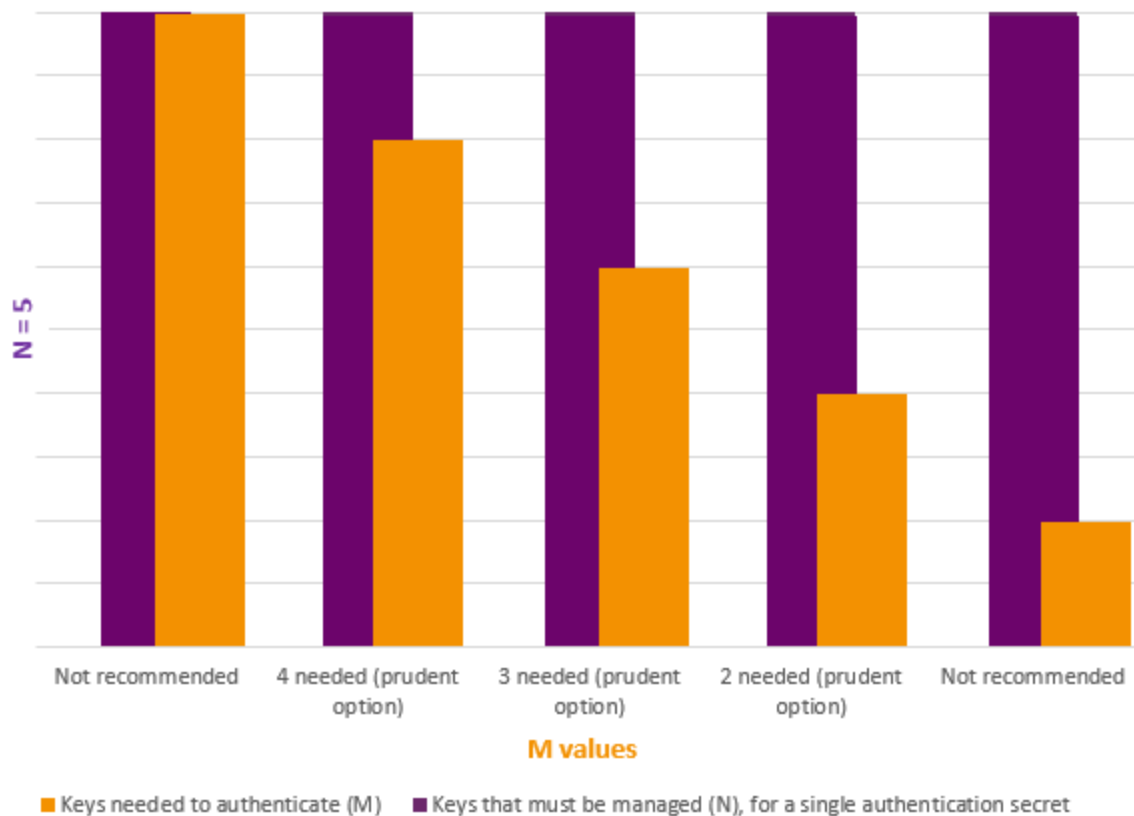
Many PED Keys for one HSM - MofN

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 an authentication secret, invoking split-secret, multi-person, access control. Choose the MofN option so that (for example) no single blue PED Key is sufficient to unlock an HSM SO role.

If you invoke MofN, two or more of a given color of PED Keys (your choice, up to a maximum of 16 splits of each secret) would then be needed to access that role or that secure function on each HSM. Distribute each split to a different person, ensuring that no one person can unlock that aspect of the HSM (HSM SO, partition SO, cloning domain, Crypto Officer and Crypto User, Remote PED, Auditor, Secure Recovery).

If you have decided that you want (in this example) three different people to be present whenever a particular role 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, beyond the initial three. You have thus specified MofN to be 3 of 5. So, if this example applied to the HSM SO, then each HSM's SO secret might be split into five components or partial secrets imprinted onto a set of five blue PED Keys, of which any three from that set can combine to reconstitute the SO secret, but never less than three.

PED Keys Needed for one authentication secret with M of N



The purple bars show $N=5$ for every choice of M , the orange bars in this example.

- $M=N$ is not recommended because it allows no scope for one of the holders to be unavailable, while still allowing you to access your HSM.
- $M=1$ is not recommended, because it is no more secure than if there were no splits of the secret - a single person can unlock the HSM role or function without oversight.
- Any choice where $N>M>1$ is prudent and useful, as it ensures oversight but allows for at least one split-holder to be unavailable, while still permitting authorized access to the HSM roles and functions.

Whether you assigned SOs to HSMs on a one-for-one or a group basis (see "One PED Key for many HSMs - grouping or reuse" on page 333 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. You can apply the above example to any of the other authentication secrets instead of (or in addition to) the SO.

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

Secrets are independent for MofN

Each HSM must have at least one application partition, in addition to the HSM administrative (SO) partition. Some SafeNet products can have multiple partitions. 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 as many as three role authentications (Partition SO, Crypto Officer, and (optionally) Crypto User) in addition to a cloning domain secret. For each of the three roles, plus the domain, you must initialize the role or secret and decide whether that role needs MofN access control, and if so, what the values of M and N should be for that role.

If you require verifiable HSM audit logs, you must initialize the HSM Auditor role (white PED Key) and decide whether that role needs MofN access control, and if so, what the values of M and N should be for that role.

If you intend to manage your PED-authenticated HSM remotely, you must initialize a Remote PED Vector (orange PED Key) and decide whether that role needs MofN access control, and if so, what the values of M and N should be for that role.

If you wish to set the HSM into Secure Transport Mode, or if you wish to require that any tamper event must be physically acknowledged and the HSM must wait until it is explicitly returned from tamper condition, then you must enable the Secure Recovery Vector (purple PED Key) and decide whether that role needs MofN access control, and if so, what the values of M and N should be for that role.

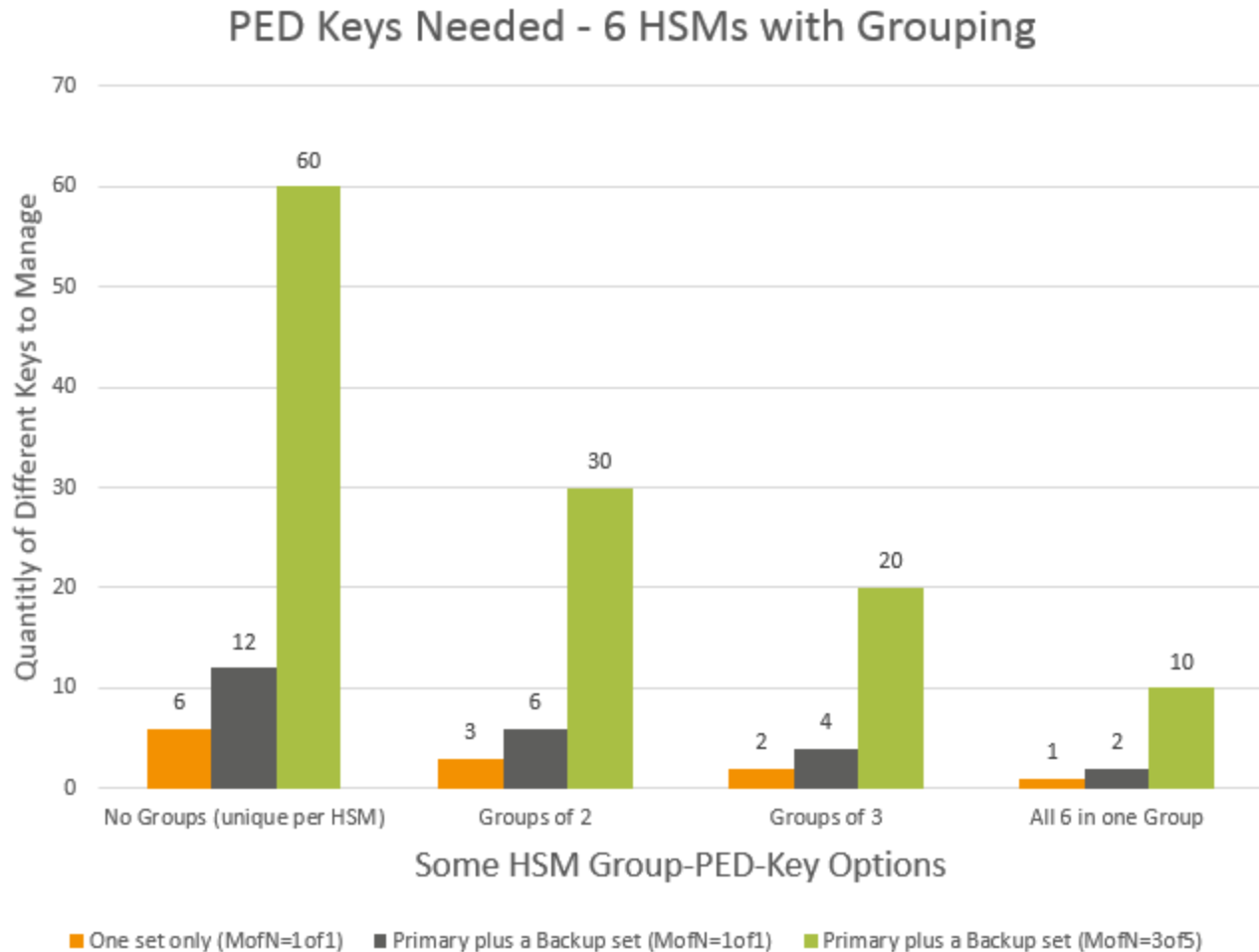
As shown, you can elect to split none of the HSM roles and secrets, some of them, or all of them. And if you do elect to impose MofN for some, or all roles and secrets, you can set different values of M and N, independently per HSM role or secret.

Combining MofN with Grouped/Unique Authentication

So, we have as many as eight different authentications on an HSM with a single application partition; nine if the application partition's cloning domain is different from the HSM SO's cloning domain. Only one of those - the purple SRK PED Key - is not subject to grouping if you have more than one HSM under your control.

Keeping in mind the need for backup sets, if you impose MofN for some or all secrets, that choice can drastically increase the number of PED Keys that must be imprinted, tracked, and managed, while the ability to group authentication secrets across some or all of your HSMs can help reduce the numbers of PED Keys in play.

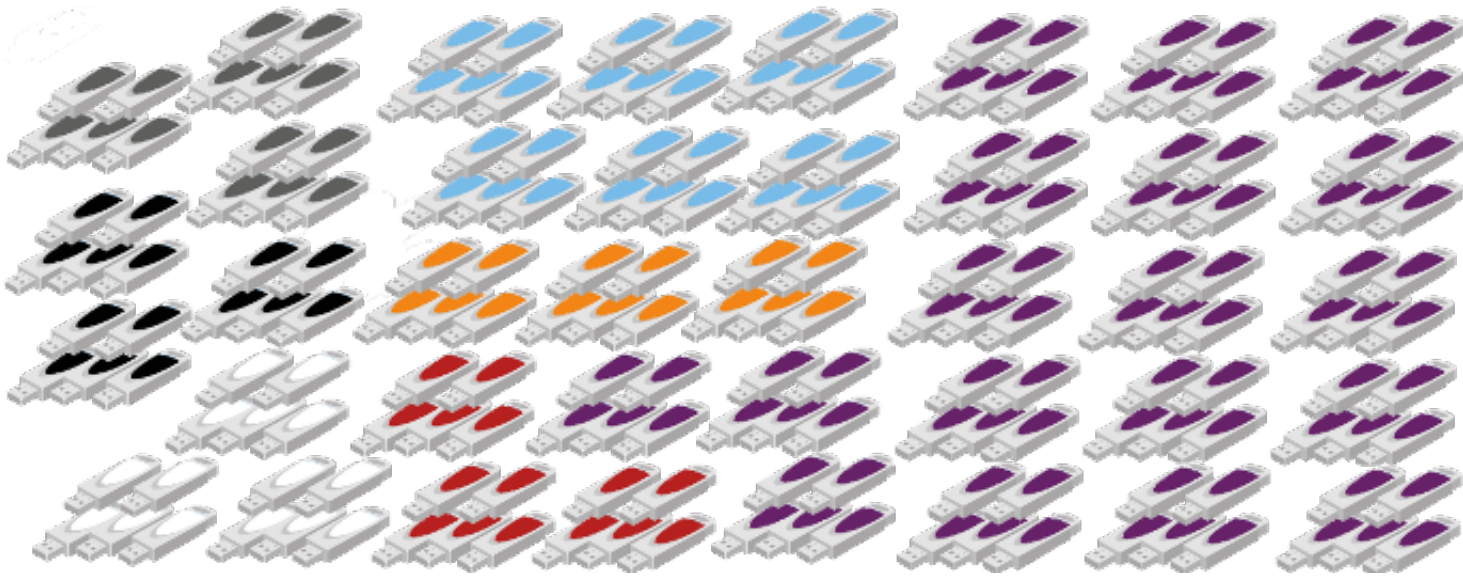
The chart below shows how the two options interact.



In the above chart, we modify the original PED Key grouping scenarios (earlier in this topic, $M=N=1$ so no MofN), by adding a requirement for MofN where the full set of keys for that secret (N) is five splits, and the number required to access that role or function of the HSM (M) is three. That covers just one secret (of the eight or nine) on six HSMs.

Here is a visual suggestion of how many physical PED Keys you would be dealing with in an example scenario where:

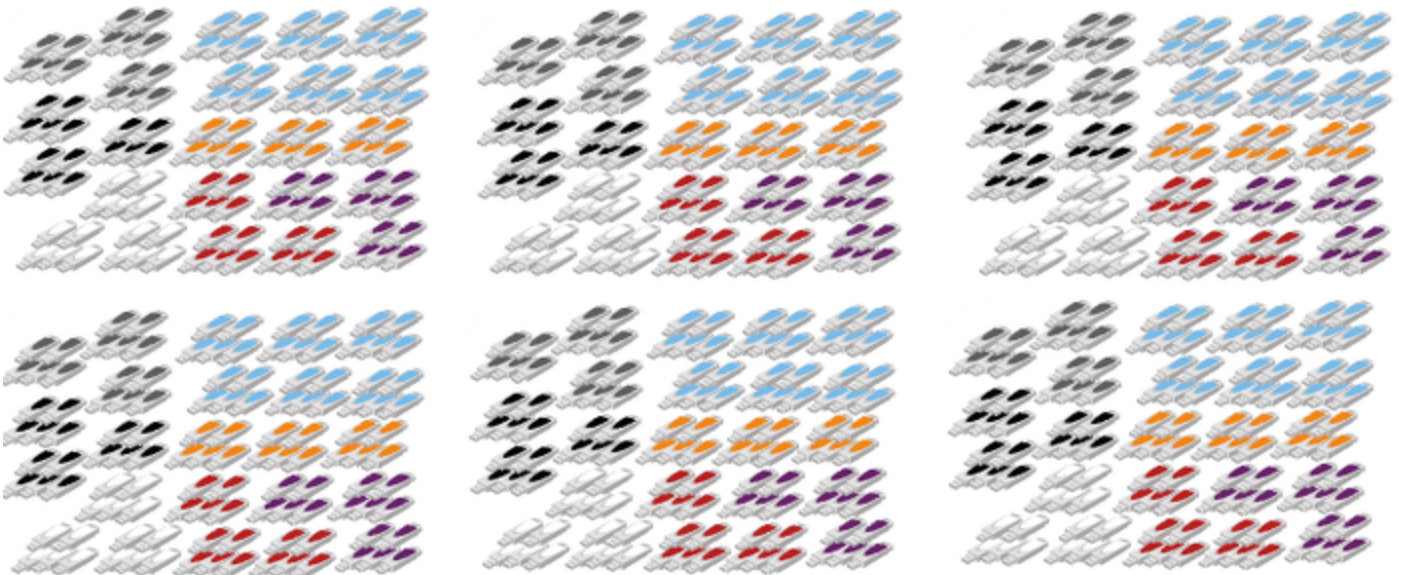
- one HSM is in play (or all secrets are grouped for several HSMs),
- you had invoked every possible role or access-controlled function on the HSM
- you had both an HSM SO and a different SO for one application partition
- you had chosen to apply MofN = 3of5 for every secret
- you were keeping an operational PED KEY set, as well as two backup sets in different locations.



Why are there so many purple keys, when we have grouped all the secrets for our six HSMs?

Recall that you can group any of the other HSM access secrets, but the purple SRK PED Key cannot be grouped. The SRK is unique to each HSM. The above illustration shows the smallest number of PED Keys you could have if you insisted on MofN at 3of5 for every secret on the six HSMs that you manage in this scenario. (The value that you chose for "M" is not important in this context. It is the value of "N" that determines how many PED Keys you must manage.) This illustrated example is a good indication that you should carefully consider whether a given HSM access secret really needs to be protected by split-secret multi-person access control (MofN greater than 1of1).

Now, look what happens if you have the six HSMs from our earlier example, and instead of grouping each HSM/secret in a single group of six, you instead elected to have no HSM grouping for any secret.



The above large number of PED Keys represents the number of physical keys you would need to manage for just six SafeNet HSMs with a single application partition per HSM where you had made these choices:

- six HSMs are in play, with no grouping of any of the HSMs for any of their secrets, so each secret is unique for every HSM,
- every possible role or access-controlled function on each HSM has been invoked (*)

- the HSM SO and the SO for the application partition on each HSM are different secrets (so a blue key set for the HSM SO and a blue key set for the partition SO... times six)
- you had chosen to apply MofN = 3of5 for every secret
- you were keeping a complete operational PED KEY set for each HSM, as well as two backup sets in different locations, for each HSM.

Granted, the illustration seems extreme, but it is the inevitable outcome if you were to make the choices that are described.

(* It is very plausible that you might have initialized different domains for the HSM and for the application partition on each HSM, in which case you could have seen an additional bunch of three sets of red keys for each HSM in the above illustration.)

The take-away observations from the examples, as you plan the deployment of your own SafeNet HSMs in your organization, should be:

- if you need a role or an access-controlled HSM function, by all means, initialize it for that HSM, but if a role or function is not needed (for example, Audit or SRK), then leave it uninitialized
- grouping of HSM secrets (re-use of keysets from other HSMs that you control, when initializing roles and functions) is good, as long as there is no countervailing security concern
- MofN is an excellent security measure, but should be invoked only when necessary, to avoid proliferation of PED Keys and the associated management burden.

Example calculation

So, with a primary keyset (for a given role or function) plus a single backup set, if all HSMs have unique secrets ("Reuse..." was chosen as "No") then you need:

$$N * G * S = T$$

where:

N is the number of MofN splits, which in this example is 5

G is the number of groups (which is 6, in our example, if all 6 HSMs have a unique secret, no grouping, or which is 1 if all 6 HSMs use the same secret for that role)

S is the number of sets (which in this case is a primary set plus one backup set, or a total of 2)

T is the Total number of PED Keys needed for just the one role or function for your six HSMs

$5 * 6 * 2 = 60$ is the largest number, with MofN at 3of5 and no grouping (no reuse of this secret across HSMs)

If all six of the HSMs use the same keyset for that role (a single group), then the number of PED Keys required to manage is greatly reduced:

$5 * 1 * 2 = 10$ in total.

That was for one secret to access a role or function of the HSM.

Calculate PED Key requirements for some or all HSM authentication secrets

The calculations above were examples for just one of the eight or nine secrets that apply to your HSM and its partitions. If more than one authentication secret is split for multi-person access control, then the number of PED Keys that you must manage grows dramatically.

Note: We strongly urge methodical handling and labeling of your imprinted PED Keys, in order to track them and to avoid mixing members from different sets.

The PED Key secret that the HSM tests, for authentication to an HSM role or function, is attempted only when it is complete.



Without MofN, the secret is complete on one PED Key. Presenting a PED Key from a wrong set, when MofN is not involved, can result in lockout of a role, or zeroization of content, depending upon which secret is attempted. For SO PED Keys, you have three tries, for other roles the default is 10 tries, but HSM or partition policies can adjust that to a lower number.

With MofN, the secret is complete only when M unique splits from the needed set, are presented and successfully combined. Presenting a PED Key from a wrong MofN set, or presenting the same split twice because members of primary and backup sets were accidentally mingled, does not allow the splits to successfully combine. Therefore that error does not increment the bad-login-attempt counter for that secret. Instead, it results in looping prompts on the PED until it gets enough of the correct splits or until the operation times out.

Maintaining Your PED Keys

Other than tracking

- where PED Keys are,
- to which HSMs they apply,
- who is in possession of, or has access to, each PED Key,

maintaining your PED Keys means

- making new copies if any suffer damage, and
- ensuring that the secrets on the keys, and any associated passwords (PED PINs, client challenge secrets) are updated in a prompt and orderly fashion, in case of suspected compromise, and in compliance with any "password-change" security policy requirements at your organization.

That last item, adherence to a policy of frequent password changes, could be the most intensive. Consider some of the examples, earlier in this topic, and how much logistical and organizational effort would be involved in a mass password change for one secret. In one example, assuming that you had all six of your HSMs in one operation center, that would be ninety PED Keys located in two different locations at your operation site (the operational set and the on-site backup set), as well as at a distant secure off-site lockup. So that would be three sets of five keys for that secret (so 15 keys for one secret for one HSM), multiplied by the six HSMs in the no-grouping scenario (so 90 keys for one secret). Now consider that one iteration of the password-change process would cover one secret (like HSM SO, blue PED Key) and would leave an additional six, or seven secrets (red, black, gray, orange, purple, white, as well as another blue key secret if your application partition has its own SO) potentially needing to be updated.

These considerations are important when you develop your authentication and security schemes around SafeNet HSMs and PED Keys.

Conclusion

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.

HSM grouping (PED Key reuse) reduces the number of PED Keys that must be managed, but puts more than one HSM at risk if a PED Key is compromised. MofN increases the number of PED Keys that must be managed, but increases the security of all affected HSM roles or functions by ensuring that no single key-holder can act unilaterally.

Using MofN

MofN 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 MofN key holders to be present for all operations. For a description of what MofN is, and how it works, refer to the Product Overview document (see "[About MofN](#)" on page 1).

Typical Practice

The typical deployment of a SafeNet Network HSM appliance is for it to be installed in a secure area of a data-center, typically near the certificate servers that it is servicing. Customers demand that this appliance is secure, but alongside that requirement they need to ensure that their processes and procedures aren't hindered by the addition of this HSM - this is the age-old security-versus- usability discussion.

The typical deployment of a SafeNet USB HSM is either attached to an application server, perhaps to serve as the root of a PKI, or attached to a SafeNet Network HSM appliance to serve in a similar capacity as part of a "PKI bundle".

The typical deployment of a SafeNet PCI-E HSM (K6) HSM is inside its application server - again, as the root of a PKI, or as the cryptographic engine to an application on that server.

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 ("[About Activation and Auto-Activation](#)" on page 268). This allows administrators of the 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 MofN 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 provide their own credentials (certificates were exchanged, to register the link) and 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 clients can no longer access the partition, regardless of their registered status and their possession of the challenge secret.

Common MofN Usage

A common customer scenario would see the HSM configured and brought into production at a data-center. This activity would need, first, the quantity M holders of blue HSM SO PED Keys, so that the HSM administrator could log in and create partitions, adjust policies, and so on. Then, in the legacy partition model, 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.

In the PPSO model of operation, you would add a second set of blue SO-split PED Keys, probably different from the HSM SO set.

Note that MofN 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 MofN for any HSM authentication secret - so only one blue SO key, and one black Crypto Officer key, one gray Crypto 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. A single person, per role, would be able to perform each function without oversight.
- You could choose to invoke MofN 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, change passwords, etc.
- You could invoke MofN 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 MofN 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 MofN sets of black PED Key splits) for every partition, or any combination of those options.

MofN and PED PINs

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 ("What is a PED PIN?" on page 296) that must be entered at the keypad when that PED Key is presented.

As with MofN, 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 MofN 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.

Revoking Means Re-initializing

Once MofN is set (either $M=1$ and $N=1$ for no multi-person access, or M and N each larger than 1 to invoke multi-person access control), that setting remains in place until the HSM or partition is zeroized and re-initialized.

So, if you decided that you wanted to stop using/requiring MofN, or that you wanted to have MofN, but with a different total number of split-keys (N) or a different minimum quantity of keys that must be presented (M) to re-construct the secret (blue, black, red, etc.), then you would need to zeroize (factory reset) and re-initialize the HSM. Or for just individual partitions, you would need to delete the partition and create a new one with the new authentication. To preserve the HSM and partition contents, you would perform backup before re-initializing the HSM, and then restore from backup afterward.

How to Add an MofN Requirement Where There Was No MofN Before

Historical Note:

On SafeNet Network HSM 4.x systems, if one HSM had MofN (using the legacy green PED Keys), and you wanted another HSM to use the same MofN splits, you had the option to clone MofN from the first HSM to the second.

Current Practice:

With SafeNet Network HSM 5 (containing the K6 HSM keycard) where MofN is a condition of each authentication secret independently, there is no provision to "clone MofN". Instead, if you wish to have two HSMs share the same MofN scheme, you must initialize one with the desired scheme, then initialize the second (and any additional) HSM and have it re-use the secret splits 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 MofN 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) 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 re-manufacturing. 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 MofN, including backups, but without the risk of accidentally mixing members of original split set and backup split sets.



Note: 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 split-containing 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 correctly observes that you are offering the same split twice.

Say, for example, that you needed an MofN set to control access to an HSM partition. You want partition access to require the presence and cooperation of 3 black PED Key holders, and you want a couple of additional splits for two alternate officers to carry, in case not all of the original three are available. So you specify MofN to be 3 of 5. You know that mishaps can occur, so you want to have a full equivalent set of black PED Key splits as backup, in case one or more of the originals is lost or destroyed. These could be stored in a secure on-site lockup, ready when needed, but requiring higher authority to release them. So you would have an original set of black PED Key splits for that partition, and a full set of duplicates.

You also want to ensure that you have a fallback in case of disaster at your operation location. So, you want a second complete backup to be held in a secure off-site lockup.

All together, you have three identical sets of five different partial secrets, for a total of 15 black PED Keys. They must be carefully labeled and handled, because you need to avoid mixing the members of different sets.

Make one big set, instead of three small sets

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.

MofN 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 MofN in your security regime, then we suggest that you not use it.

If your security policy demands that you use MofN 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

These 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 SafeNet 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 SafeNet 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 `lunash` or at the client when you are about to perform an action that could alter an HSM, a token or its contents. You might consider a label consisting of a part (perhaps a word) that identifies the domain to which the token 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 HSM Servers 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 Server, 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.

Do not permanently obscure the appliance serial number. You would want it visible in the unlikely event that you ever needed to contact SafeNet for assistance.

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, 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 MofN option, which adds several, visually-similar keys to the mix. It might be useful to identify the following:

- Which keys (blue, black, red) are associated with which HSMs or Partitions or Clusters).
- 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 MofN 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 MofN status of keys would be useful once the keys and tokens are in operation (or in backup safe storage), or whether your security requirements would prohibit such tags or markings.

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 MofN 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 SafeNet 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 SafeNet 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.

Updating PED Key for a Backup Token

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

Performing an HSM Backup or a Partition Backup will initialize the token 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 token) ,

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 tokens, or on a SafeNet HSM.

Frequently Asked Questions

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

How should SafeNet PED Keys(*) be stored? (*Model iKey 1000 for use with SafeNet 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 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 SafeNet 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 SafeNet HSMs and SafeNet 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, SafeNet 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 SafeNet 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 November 2014) 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 re-manufacture. 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 re-manufacture 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.

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

- ["Performance Overview" below](#)
- ["HSM Information Monitor" on the next page](#)
- ["Performance and the PE1746Enabled Setting" on page 358](#)
- ["Frequently Asked Questions" on page 358](#)

Performance Overview

SafeNet Network HSM 5.x/6.x has a newer generation internal HSM; for discussion purposes, SafeNet refers to this HSM as K6. SafeNet refers to the HSM inside SafeNet Network HSM 4.x as K5. Both K5 and K6 rely on application-specific integrated circuits to accelerate cryptographic operations within the HSM. Each generation uses different ASICs. These ASICs use multiple "engines" – analogous to the processor inside a computer having multiple central processing units – to spread load and thereby increase performance.

With SafeNet Network HSM 4.x, a client application needs to create about 20 threads to achieve maximum RSA signing performance based on 1024-bit RSA operations. At this number of threads, within the K5, the ASIC achieves optimal distribution of cryptographic operations across its multiple engines. The ASIC within K6 has a different number of engines and a different algorithm for distributing load. To achieve maximum performance with SafeNet Network HSM 5.x or 6.x, a client application needs to create about 50 threads. With refinements made for SafeNet Network HSM 5.2, this number is now about 30 threads, and carries through for SafeNet 6.x.

Published performance figures for SafeNet Network HSM generally reflect repeated single operations against a single object that is imported or looked up one time before all the operations are performed. This is the most advantageous situation, under the best conditions to yield the highest attainable speed with the equipment. All manufacturers take the same approach.

"Real life" performance figures are often lower because of additional overhead, such as where an object must be fetched before each operation, or where the current task switches constantly from one operation type to another (example sign-and-verify in combination).

If you are using (say) the supplied multitoken tool in a lab setting, note that it defaults to a packet size of 1 kilobyte for symmetric encrypt/decrypt operations, a modest size that imposes a significant overhead. To obtain performance closer to "real life" for your situation, the test packet size should be modified to match the sizes that you expect to see in your intended application. For example, a packet size on the order of 256 bits for credit card numbers versus 64 kilobytes and larger for high-throughput encryption could show significantly different performance.

When HA is considered (two or more HSMs in a redundant group), further overhead is introduced in order to replicate/synchronize across all members of the group. Therefore, the type of operation - whether it requires a single initial replication before a large volume of operations against a static object, or whether it requires a new replication before each single operation - can have a very significant impact on performance.

HA Performance

For repetitive operations, like a high volume of signings using the same key, an HA group can expand SafeNet Network HSM performance in linear fashion as HA group members are added. HA groups of 16 members have undergone long-term, full-throttle testing, with excellent results.

Do keep in mind that simply adding more and more SafeNet Network HSM appliances to an HA group is not an infallible recipe for endless performance improvement. For best overall performance, all HA group members should be driven near their individual performance "sweet spot", which for SafeNet Network HSM 5.2 and later is around 30 simultaneous threads per HSM. If you assemble an HA group that is considerably larger than your server(s) can drive, then you might not achieve full performance from all.

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

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 do the following:

- 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.

For lunacm, see ["hsm monitor" on page 1](#) in the *LunaSH Reference Guide*.

Performance and the PE1746Enabled Setting

The K6-based HSMs include the SafeXcel 1746 security co-processor, which is used to offload packet processing and crypto computations from the host processor. Use of the SafeXcel 1746 security co-processor can affect performance, and is therefore optional. When enabled, the SafeXcel 1746 security co-processor improves application bulk performance, at the expense of small-packet performance. When disabled, small-packet performance is improved, at the expense of application bulk performance. Data packets less than 1Kb in size are considered small.

You can enable or disable the SafeXcel 1746 security co-processor via the **PE1746Enabled** statement in the **Chrystoki.conf** file (Linux and UNIX) or the **crystoki.ini** file (Windows). The SafeXcel 1746 security co-processor is disabled (0) by default.



Note: K6-based HSMs have a limit of 1000 contexts for SafeXcel 1746 operations, which is a consideration when many client threads are involved, and depends upon the number of concurrent threads.

To enable or disable the SafeXcel 1746 security co-processor

1. Login to your SafeNet HSM client workstation as an administrator.
2. Open the **Chrystoki.conf** (Linux and UNIX) or **crystoki.ini** (Windows) file, as relevant, for editing. The **PE1746Enabled** statement is located in the Misc section of the file, for example:

```
Misc = {
PE1746Enabled = 1;
reconnAtt = 50;
logLen = 262144;
haLog = /usr/safenet/lunaclient/bin/;
}
```

3. Set the value for **PE1746Enabled** as required. Set to 1 to enable. Set to 0 to disable.

Effect on HA

The **PE1746Enabled** setting can affect HA. See "Performance" on page 211 for more information.

Resetting the Internal SafeNet Network HSM PE1746Enabled Setting Following an Upgrade

Because of the effect on some operations, it can happen that a large update to SafeNet Network HSM can fail verification if PE1746Enabled= 0 in the SafeNet Network HSM's internal configuration settings. A patch is available to force PE1746Enabled= 1 on the appliance.

Frequently Asked Questions

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

Can I buy a SafeNet Network HSM 1700 and later upgrade it to SafeNet Network HSM 7000?

No. The SafeNet Network HSM 1700 appliance comes with a single power supply, and you can purchase a second PS, remove the blanking plate and install the new PS into the empty slot for the same power-supply redundancy as SafeNet Network HSM 7000, but you cannot increase the SafeNet Network HSM 1700 performance. If you need the performance, buy a SafeNet Network HSM 7000.

Note that you could buy and use the less-expensive SafeNet Network HSM 1700 for your testing, development, or integration, and then purchase SafeNet Network HSM 7000 appliances for your operational deployments where higher performance is desired. Other than performance, the two appliance models are functionally identical. An integration or application that works with one will work with the other, with no adjustment needed.

Note that the main difference (see next question) is performance using asymmetric operations. If your primary cryptographic activity is key-generation of any kind, or involves mostly symmetric operation, then SafeNet Network HSM 1700 should suit your needs at a smaller investment.

Can you highlight the relative performance figures?

Here is a quick summary of some excerpts from the performance testing:

- Key Generations – same performance for both
- Symmetric operations – same performance for both
- Asymmetric operations – check with your SafeNet Sales representative, but here are some samples.
 - RSA 1024: 7000 signings/second vs 1700 signings/second
 - RSA 2048: 1200 vs 350
 - RSA 4096: 160 vs 50
 - ECC P-256: 1000 vs 490

How can I achieve the kinds of performance numbers you quote?

We provide repeatable numbers that you could achieve if you tested in an environment similar to our test-lab environment. This method provides numbers that you can compare against numbers from any of our competitors. In general, that means automated scripting to perform a given operation with a "standard" keysize, or standard input parameters, and ensuring that no other operation or latency is allowed to intrude - an isolated high-speed network with no other activity.

Operation outside a controlled laboratory environment is messy and sometimes unpredictable, with many variables that could affect testing if we did not control the parameters and conditions as tightly as possible. Therefore, all manufacturers' test numbers tend to be the best you can get under ideal conditions.

So, for example, we use the RSA 1024-bit key as the standard for performance of asymmetric sign and verify operations, because that is what the industry uses as a common basis of comparison. That remains true even though the 1024 key size is now generally considered too small for modern operational use, and most applications would tend to use 2048 key sizes (at least that was the case when this was written in 2013).

As well, we bombard the HSM with at least 30 threads simultaneously performing that simple test operation (this is down from a previously required 50 threads due to refinements in SafeNet Network HSM 5.2). Any fewer might fail to exercise the HSM to its fullest. Significantly more threads would not increase the performance numbers, so we work in the "sweet spot" and we suggest that you do as well if performance is your greatest concern.

In operational, real-world situations, your clients are likely to have other responsibilities, or might make other requests of the HSM - for example, a fresh asymmetric key generation before each asymmetric signing operation would slow the sign/verify performance down to key-gen speeds... orders of magnitude slower than simple, repetitive signing that reuses a single key. Network latency and other factors also serve to degrade performance in non-laboratory operation.

Do you have any additional advice on how to interpret performance numbers? We're trying to match against a set of performance requirements that are stated as "signings per millisecond".

Remember that our numbers are not based upon sequential access, but are based upon an optimal number of concurrent threads accessing the HSM. That optimal number differs from one HSM to the next. Best performance for the legacy SafeNet CA4 is achieved from roughly 4 - 8 threads with a slight drop-off after that due to the overhead of context switching. On SafeNet Network HSM 5.0 and 5.1, the optimal number of threads has been in the range of 50, but refinements in SafeNet Network HSM 5.2 have brought the optimal number of concurrent threads down to 30.

Command latency (the time required for any one command to complete) is not a direct inverse of TPS, and can be dependent on several things including the number of threads, the network latency, and the interleaving of different command types to the HSM. (That is, ideal signing performance can be achieved only if the HSM is processing only signing requests.) As the number of threads increases (and thus the corresponding TPS) the latency increases as well. So you might be seeing what appears to be 6 responses per millisecond, but the round-trip latency of any one of those commands might be as high as 500 ms (for example).

This is partially why, when you observe numbers from our tools, the numbers are quite variable for the first several seconds, and then settle around stable values as the testing proceeds.

Performance requirements should state both the total throughput and the maximum latency that a command must execute, and the ideal number of threads should be adjusted accordingly.

We expect to generate millions of keys per year. What is the expected number of read/write operations that your HSM memory can perform before the solid-state memory begins to fail?

You are thinking of the HSM's flash memory, which would be used to store token objects. By their nature, those do not frequently change.

Your "key factory" application (generating keys that are pushed out to external devices like smart cards) should be generating your short-lived keys as session objects, rather than as token objects. Session objects do not use the flash memory at all - they are created and exist in the HSM's RAM only, which can perform virtually unlimited read/write operations.

In the "key factory" scenario, we need to generate approximately 30 ECC P224 keys/second. How many SafeNet Enterprise HSMs will we require?

SafeNet Network HSM 5.2, and later, can do about 35 ECC P224 keys per second. You would then need one appliance for performance reasons, if you could count on steady demand with no peaks. Consider having a backup as well.

In the situation where you might encounter bursts of key generation traffic, prudence suggests one operational SafeNet Network HSM, one in standby to accommodate the burst traffic, and a third unit for backup.

Public Key Infrastructure (PKI) and Removable HSMs

This chapter describes PKI in the context of SafeNet HSMs. It contains the following sections:

- "PKI with SafeNet Network HSM" below
- "Using SafeNet USB HSM or Token-format HSM with SafeNet Network HSM Appliance" on page 363
- "Card Reader (SafeNet DOCK 2) and Token-style HSMs" on page 366
- "Frequently Asked Questions" on page 368

PKI with SafeNet Network HSM

The PKI feature with SafeNet Network HSM is summarized as follows:

- Legacy SafeNet PCM token HSMs can be configured as PKI devices via SafeNet Dock 2 (an external, USB-connected SafeNet card reader).
- SafeNet USB HSMs can be connected to the SafeNet Network HSM USB port and configured as a PKI device.
- Each PKI device can support only one partition.
- One SafeNet Network HSM can support up to six PCM token HSMs (via three SafeNet Dock 2 readers), or three SafeNet USB HSMs, or a mix of both (limited to three USB connections to the appliance).

What to Do

If you are an end-user of SafeNet HSM products, then it is assumed that you are using your SafeNet HSM in conjunction with a third-party application that is HSM-aware. Simply follow the instructions and procedures associated with that application, once you have installed the SafeNet HSM and configured it (described in the Installation Guide and Configuration Guide).

If you are a developer or integrator of applications, then refer to the Software Development Kit Guide, along with the "Extensions to PKCS # 11" (SafeNet's augmentation of the PKCS # 11 standard API), and in particular to the token pki commands in the Reference section of this Help.

Special commands are provided under the token pki menu to perform HSM management operations on the removable HSMs (SafeNet tokens or SafeNet USB HSMs). Briefly, to make use of SafeNet tokens and SafeNet USB HSMs with SafeNet Network HSM, you need to use:

- similar to initializing the onboard SafeNet Network HSM, but prepares the removable token to be used in this context
- make the named token available to the SafeNet Network HSM appliance as an additional PKCS#11 slot (like an additional, removable HSM Partition)
- make the deployed token/slot available/accessible to Clients

- generate or clone to populate the token with the necessary keys, certs, etc.

The is used to make the inserted, deployed token unavailable, such as when preparing to remove it. The remaining commands, under token pki are for general management of the tokens, and are similar to equivalent HSM and Partition commands.

Lunash "token" command set provides 16 commands to administer the external PKI HSM (SafeNet USB HSM). You need just two of those "token" commands, plus one "client" command to make the PKI HSM ready to use, as follows:

1. Pre-deploy the external HSM, to prepare it. Type:

```
[mylunaSA] lunash:>token pki predeploy -l G5Pki -serial 475289
```

```
[mylunaSA] lunash:>token pki predeploy -l G5Pki -serial 475289
```

```
Please type "proceed" to continue, anything else to abort: proceed
```

```
*****
*
*      About to factory Reset the HSM      *
*
*****
*****
*
*      About to initialize the HSM          *
*      Please pay attention to the PED      *
*
*****
```

```
Do you want to use FIPS-approved algorithms and key strengths only (yes or no)? Yes
```

```
*****
*
*      About to change the HSM FIPS policy  *
*      Please pay attention to the PED      *
*
*****
*****
*
*      About to create a partition on the HSM *
*      Please pay attention to the PED      *
*
*****
*****
*
*      About to set the partition policies   *
*      Please pay attention to the PED      *
*
*****
*****
*
*      About to create a partition challenge *
*      and activate the partition.          *
*      Please pay attention to the PED      *
*      Please write down the PED secret!   *
*
*****
Success predeploying the token!!
```

Command Result : 0 (Success)

[mylunaSA] lunash:>

- Now deploy the pre-deployed HSM to make that HSM available to the SafeNet Network HSM as another application partition or PKCS#11 slot. Type :

[mylunaSA] lunash:>token pki deploy -label G5Pki -serial 475289

```
*****
*
*   About to activate the token for testing. *
*   Please pay attention to the PED         *
*                                           *
*****
Please enter the current user challenge:
Success deploying token StellaG5Pki with serial num 475289 !
```

Command Result : 0 (Success)

[StellaSA2] lunash:>token pki listDeployed

Label	Serial Num
-----	-----
StellaG5Pki	475289

Command Result : 0 (Success)

[StellaSA2] lunash:>client assignPartition -partition StellaG5Pki -c StellaLap

'client assignPartition' successful.

Command Result : 0 (Success)

HA

The SafeNet Network HSM's HA (high availability) feature, when implemented for PCM tokens or SafeNet USB HSMs must be used only across multiple SafeNet Network HSM appliances. NEVER allow multiple SafeNet PCM tokens or SafeNet USB HSMs to be placed in an HA configuration on a single SafeNet Network HSM appliance. This is similar to the requirement to not include two partitions of the same HSM in a single HA group.

Using SafeNet USB HSM or Token-format HSM with SafeNet Network HSM Appliance

Traditionally, Public Key Infrastructure (PKI) with SafeNet HSMs has been implemented using removable token-style (PCMCIA format) HSMs securely connected to a local workstation via a card reader. The portable HSM contained the PKI root certificate, and was inserted, read, updated, etc., as needed, then removed and returned to safe storage. This was a high-security, low-volume/low-speed environment and requirement.

This differed from the transaction-security world where HSMs needed to be network-available in order to perform and accelerate high volumes of secure transactions.

When those two applications began to converge, the SafeNet Network HSM, with its model of large, fast, network-connected HSM providing multiple virtual-HSM (Partition) workspaces, was adapted to support the addition of token-format PKI HSMs (such as SafeNet PCM or SafeNet CA4).

External HSMs (Token-style and G5 style)

You can connect a SafeNet DOCK2 card reader for limited use with SafeNet Backup tokens (legacy G4 PCMCIA removable token-format HSMs). The removable-token backup HSM was used to backup legacy SafeNet Network 4.x HSMs and can be connected to SafeNet Network HSM 5.x or 6.x to restore the legacy key material as part of a one-way migration.

You can connect the more modern SafeNet USB HSM as an externally connected PKI slot, for use in the PKI Bundle option. Some customers use this arrangement to hold a root CA. The following caveats apply:

- The **token backup** commands can see and manage only the backup device, and not PKI devices.
- The **token pki** commands can see and manage only the PKI devices, and not backup devices.
- The PKI device must use PED authentication only, to be deployed.
- The **token pki update** commands update the capability and firmware for PKI devices.
- The process to move keys off G4 token HSMs (SafeNet CA4) is to migrate the keys to a K6 HSM (either the K6 inside SafeNet Network HSM, or the standalone K6 (SafeNet PCI-E HSM inside a host computer)) and then to SafeNet USB HSM. Cloning between G4 and G5 devices is not supported.



CAUTION: Migration is not supported to firmware 6.22.0. Migrate first to an HSM at a firmware version older than 6.22.0, and then update the HSM firmware to version 6.22.0 or newer.



CAUTION: Beginning with SafeNet HSM 6, we do not support PKI bundle using removable PCMCIA token HSMs (SafeNet CA4) and the SafeNet DOCK 2 reader. The SafeNet DOCK 2 reader is supported only for migration. If you need the PKI bundle function from removable tokens, do not upgrade.



Note: PPSO is not supported for the PKI-bundle configuration using SafeNet USB HSM. There is no provision to apply PPSO capability via SafeNet Network HSM to the externally connected SafeNet USB HSM. If the SafeNet USB HSM was removed to a host computer and updated to firmware 6.22.0 and had the PPSO capability applied (destructive operation), then returned to the SafeNet Network HSM to resume PKI-bundle operation, the interface has no provision to create a PPSO partition in the external HSM. Rather, a legacy-style partition would be created for PKI-bundle operation.

Constraints

To use an external PKI HSM directly with SafeNet Network HSM 5 requires a SafeNet USB HSM, or a SafeNet DOCK2 reader with SafeNet CA4 token-style HSM at firmware 4.8.7 or later.

Whether you are using the onboard HSM or not, in order to use a SafeNet Network HSM for PKI bundle operations (using Luna/HSM CA4 or Luna/HSM PCM tokens in the appliance's card-reader) you **must** at least **initialize** the **onboard (K6) HSM** in order to use the connected HSMs. Any further preparation of the onboard HSM depends on how

(or if) you intend to make use of it, but having the main HSM initialized before you attempt operations with PKI HSMs connected to it is a minimum requirement.

PKI and HA

You can combine the PKI bundle configuration (a SafeNet USB HSM, or a SafeNet DOCK2 with inserted SafeNet CA4, connected to your SafeNet Network HSM appliance) with the HA grouping functionality. That is, PKI can be part of HA redundancy and load balancing. However, by design, we do not support the assigning of two or more devices from the same SafeNet Network HSM to one HA group. That is:

- while SafeNet Network HSM supports multiple HSM partitions, you cannot combine two or more partitions from one SafeNet Network HSM into an HA group, and
- while you can attach a SafeNet USB HSM or a SafeNet CA4 token HSM to a SafeNet Network HSM, you cannot combine two (or more) HSMs or partitions, associated with a single SafeNet Network HSM, into a single HA group.

In either case, that sort of arrangement would allow the SafeNet Network HSM to become a potential single-point-of-failure, which defeats HA's redundancy.

Instead, if you have multiple SafeNet USB HSMs or SafeNet CA4 token HSMs that you wish to use in PKI bundling with SafeNet Network HSM, then you should connect each SafeNet USB HSM or SafeNet CA4 HSM to a separate SafeNet Network HSM. You should not attempt to include more than one SafeNet Network HSM partition, or a partition and an externally connected HSM, in a single HA group. The HA logic recognizes HA member slots from different NTLA/NTLS links, only. This is by design.

Slot Enumeration

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

```
bash-3.2# ./vtl listslot
Number of slots: 11
The following slots were found:
```

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



Note: The deploy/undeploy of a PKI device increments/decrements the SafeNet Network HSM client slot enumeration list (slots appear or disappear from the list, and the slot numbers adjust for the change). When the PKI slot is temporarily not available (e.g., due to NTLS stop, unplugging of LAN/USB cable, power off, etc.), the slot list does not shift.



Note: If you attempt to perform actions (such as deployment) that require PED operations, against a token/HSM, while other applications are accessing either the onboard HSM or another token in your appliance, then the PED-requiring operations might be noticeably slow. In general, try to reserve such maintenance operations for times when clients are not accessing the HSM or other token. The possible slowness is merely inconvenient and does no harm.

See also "Card Reader (SafeNet DOCK 2) and Token-style HSMs" below.

Contact SafeNet Technical Support – e-mail: support@safenet-inc.com or phone 800-545-6608 (+1 410-931-7520 International) for the relevant Key Migration document, which includes explicit instructions to migrate your cryptographic objects between different types of SafeNet HSM (generally from legacy models to current models of HSM).

Card Reader (SafeNet DOCK 2) and Token-style HSMs

The card reader sold for use with SafeNet products (PKI) is the SafeNet DOCK 2.



Uses with SafeNet Network HSM 6 are:

- for migration from earlier backups or PKI tokens
- for current (limited) use of legacy PKI tokens (SafeNet CA4) with SafeNet Network HSM.

External HSMs (Token-style and G5 style)

You can connect a SafeNet DOCK2 card reader for limited use with SafeNet Backup tokens (legacy G4 PCMCIA removable token-format HSMs). The removable-token backup HSM was used to backup legacy SafeNet Network 4.x HSMs and can be connected to SafeNet Network HSM 5.x or 6.x to restore the legacy key material as part of a one-way migration.

You can connect the more modern SafeNet USB HSM as an externally connected PKI slot, for use in the PKI Bundle option. Some customers use this arrangement to hold a root CA. The following caveats apply:

- The **token backup** commands can see and manage only the backup device, and not PKI devices.
- The **token pki** commands can see and manage only the PKI devices, and not backup devices.
- The PKI device must use PED authentication only, to be deployed.
- The **token pki update** commands update the capability and firmware for PKI devices.
- The process to move keys off G4 token HSMs (SafeNet CA4) is to migrate the keys to a K6 HSM (either the K6 inside SafeNet Network HSM, or the standalone K6 (SafeNet PCI-E HSM inside a host computer)) and then to SafeNet USB HSM. Cloning between G4 and G5 devices is not supported.



CAUTION: Migration is not supported to firmware 6.22.0. Migrate first to an HSM at a firmware version older than 6.22.0, and then update the HSM firmware to version 6.22.0 or newer.



CAUTION: Beginning with SafeNet HSM 6, we do not support PKI bundle using removable PCMCIA token HSMs (SafeNet CA4) and the SafeNet DOCK 2 reader. The SafeNet DOCK 2 reader is supported only for migration. If you need the PKI bundle function from removable tokens, do not upgrade.

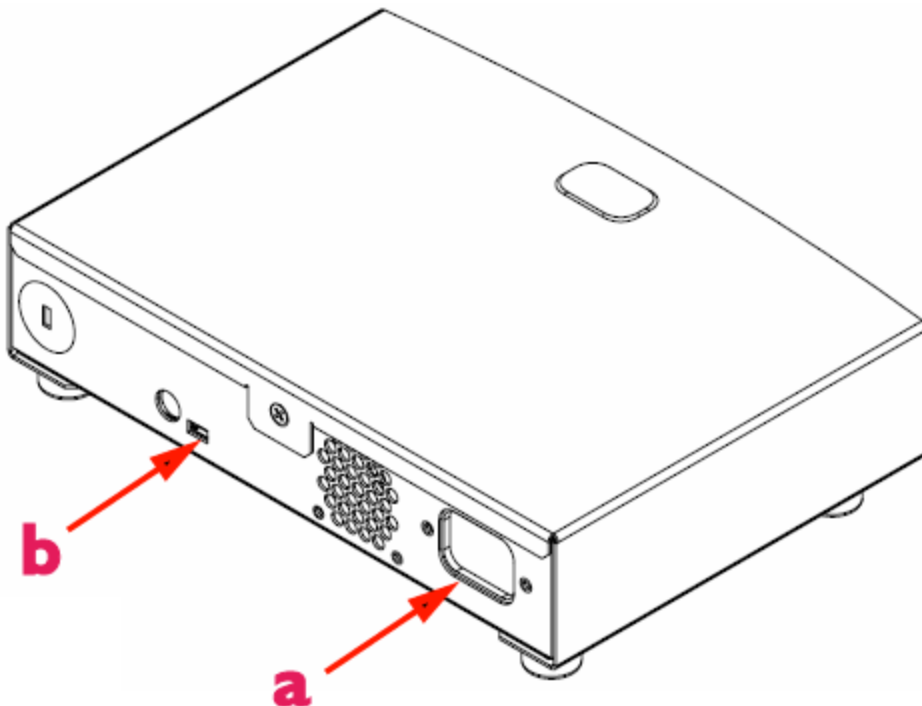


Note: PPSO is not supported for the PKI-bundle configuration using SafeNet USB HSM. There is no provision to apply PPSO capability via SafeNet Network HSM to the externally connected SafeNet USB HSM. If the SafeNet USB HSM was removed to a host computer and updated to firmware 6.22.0 and had the PPSO capability applied (destructive operation), then returned to the SafeNet Network HSM to resume PKI-bundle operation, the interface has no provision to create a PPSO partition in the external HSM. Rather, a legacy-style partition would be created for PKI-bundle operation.

Do not install SafeNet client software on the same system as legacy SafeNet CA³, SafeNet CA4, SafeNet PCM, or SafeNet PCI software. The software is intended for modern/current SafeNet HSMs, SafeNet Network HSM, SafeNet PCI-E HSM, SafeNet USB HSM, SafeNet (Remote) Backup HSM.

Connect the SafeNet DOCK2 card reader:

- a) to the AC main power, and
- b) via supplied USB cable to the USB port of your SafeNet Network HSM 5.x.



If power is disconnected for any reason, you might need to restart your application.

The SafeNet PKI Bundle feature supports PED-authenticated PKI HSMs only (SafeNet CA4 for legacy, and SafeNet USB HSM for modern). Use of password-authenticated PKI tokens is not supported. There is no "pass-through" of PED data and commands from SafeNet Network HSM, so your SafeNet DOCK2 (or SafeNet USB HSM) must have its own SafeNet PED connected directly.



Your SafeNet Network HSM needs its own SafeNet PED.

SafeNet Network HSM can be served by a locally-connected PED, if the administrator is located near the appliance, or SafeNet Network HSM can be served by Remote PED, but SafeNet DOCK2 and any inserted token HSMs require a PED to be connected directly and locally to the reader - use of Remote PED to serve an external HSM (such as SafeNet USB HSM, SafeNet Backup HSM, or SafeNet CA4) connected to SafeNet Network HSM is not supported.

See also [PKI - Using an external HSM with SafeNet Network HSM Appliance](#).

Contact SafeNet Technical Support -- e-mail: support@safenet-inc.com or phone 800-545-6608 (+1 410-931-7520 International) for the relevant Key Migration document, which includes explicit instructions to migrate your cryptographic objects between different types of SafeNet HSM (generally from legacy models to current models of HSM).

Frequently Asked Questions

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

We operate a Managed PKI and must satisfy our auditors that the root and intermediate keys and certs are protected according to an accepted standard, including when cloned/backed-up.

We have documented procedures for cloning or backup/restore, and for migration from legacy HSMs to current HSMs, but the procedures are only to ensure that the operations complete successfully. Security of private keys is enforced by

the HSM(s) and does not rely on procedure.

The encryption key is either 3-key TDES or AES 256, depending on the HSM firmware version, which itself is afforded the same high level of protection as a CA signing private key. The encryption key is derived using the data from the Red PED Key (48 bytes of HSM-generated random data) along with source and target HSM random nonces that are exchanged using RSA 2048 bit encryption. Both the source and target HSMs must be legitimate SafeNet HSMs and their RSA certificates (used to exchange encrypted nonces) are signed by the SafeNet manufacturing PKI when the devices are manufactured.

15

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" below
- "Remote PED Architecture" on page 376
- "Remote PED and pedclient and pedserver" on page 377
- "Configuring Remote PED" on page 379
- "Using the Remote PED Feature" on page 388
- "Troubleshooting Remote PED" on page 395

About Remote PED

The Remote PED (SafeNet PED with Remote Capability) allows you 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 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
- a SafeNet PED 2.4.0-3 or greater, **with** the Remote PED feature installed, (which has the capability to operate in Local PED or Remote PED mode, as needed).



Note: Not every PED 2.4.0 includes the Remote PED feature. That PED capability must be ordered specifically and factory installed.

- an orange RemotePED PED Key, which provides the authentication for the Remote PED connection between the workstation computer (with SafeNet PED 2 connected and PEDServer running) and the remotely located SafeNet HSM with the PEDclient running on the HSM's host.

Term	Meaning
Remote PED	A SafeNet PED, with Remote capability, connected, powered on, and set to Remote mode.
RPV	Remote PED Vector - a randomly generated, encrypted value used to authenticate between a Remote PED (via PedServer) and a distant SafeNet HSM (PEDclient).

Term	Meaning
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 SafeNet HSM).
PEDClient	The PED Client program. For a SafeNet Network HSM appliance, PEDclient is embedded. For SafeNet PCI-E HSM, SafeNet USB HSM, or SafeNet Backup HSM, PEDclient must be installed on the HSM's host computer. The PED client 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?

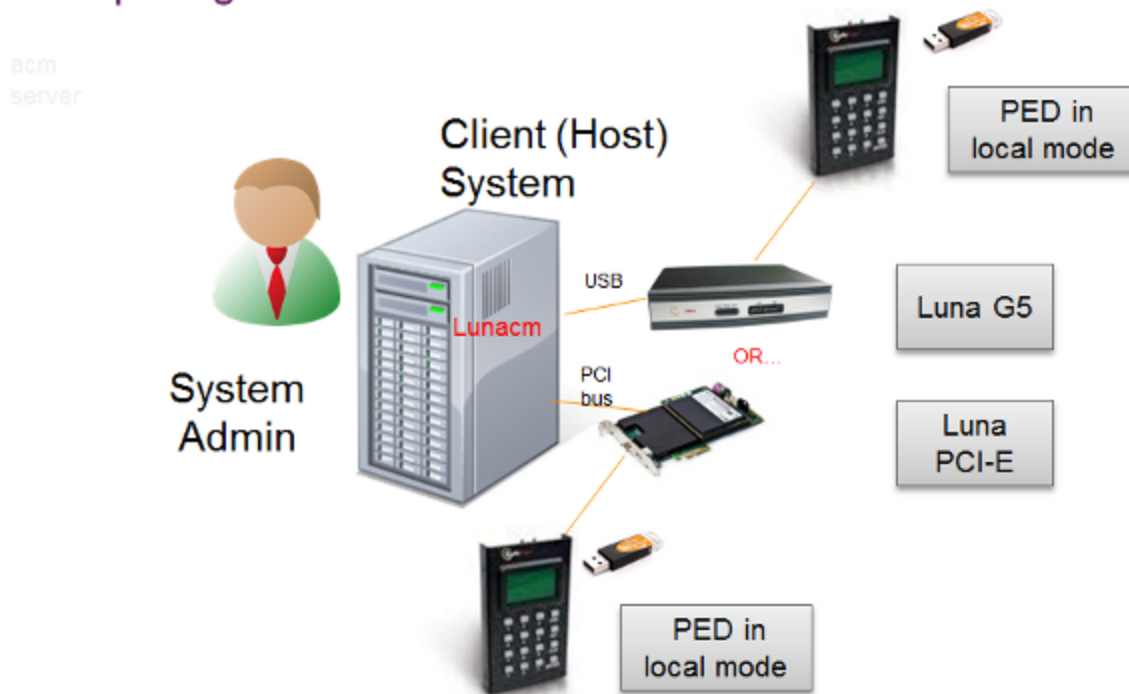
You want to locate your operational HSM hosts at remote locations or multiple locations around the city, country, world, and be able to administer them fully, from one location, without need for site visits and without carrying of PED Keys through unsecured areas.

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, 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 following 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 locally connected PED. 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.

Preparing for Remote PED



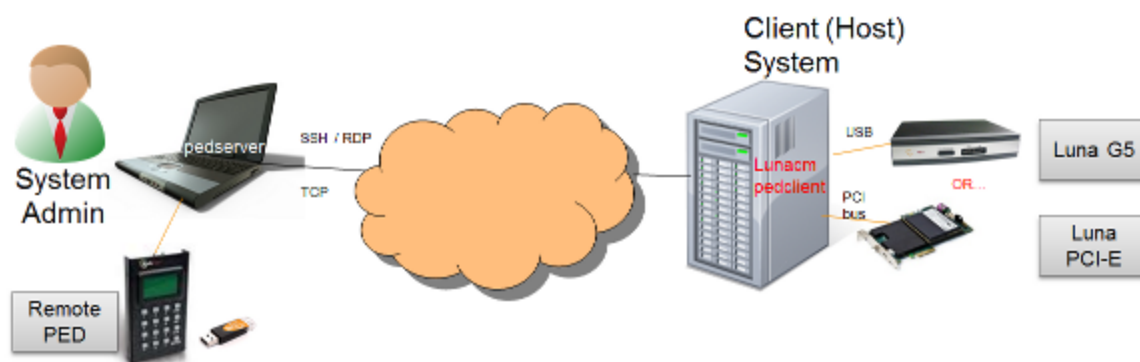
The HSM is then shipped and installed at its remote location.

At your administrative location, a workstation is configured with special (PedServer) software, and a SafeNet 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 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.

Using Remote PED



One-to-One Remote PED Connections

A SafeNet HSM running the PED client can establish a Remote PED connection with any workstation that meets the following criteria:

- is running PEDserver.exe
- has a suitable Remote PED connected
- has the correct PED Keys (including the orange key) for that HSM.

The SafeNet 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 SafeNet HSM.'

Similarly, a given workstation can enter into a Remote PED connection with any SafeNet HSM with PEDClient, or any SafeNet HSM, 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 SafeNet HSMs.

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

Priority and Lockout

A Remote PED connection is always initiated from the SafeNet 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 PedServer.exe, and there is no provision to use that program to connect to the Remote PED client on a SafeNet HSM-attached host computer, or a SafeNet Network HSM appliance. Nevertheless, you could open an SSH window on that same workstation "A" (or on any other computer), connect to the SafeNet Network HSM appliance or the SafeNet HSM host computer, log in, and tell the host to initiate a Remote PED connection with workstation "A". The appliance or HSM host computer does not care which computer runs the SSH (or local serial) connection to its admin interface. The function of a communication connection for SafeNet shell [lush] on SafeNet Network HSM, or for a computer hosting a SafeNet PCI-E HSM or SafeNet USB HSM, is completely separate from the function of a communication connection for Remote PED operation.

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** (LunaCM) command or **hsm ped connect** (LunaSH) command, one of the following two things would happen:

- the remote PED connect command would begin executing, but would pause while the local-PED operation (started in the other command session) was in progress, and resume when the local-PED operation terminated
- the remote PED connect command would begin executing, but would pause while the local-PED operation was in progress, and 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 SafeNet PED is connected directly to the HSM. Timeouts are governed by the configuration of the appliance or host computer and the HSM and are not generally modifiable.

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 configuration 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 SafeNet Network HSM appliance or on an HSM host computer, are not aware of each other's 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 appliance or HSM host that the connection had been open too long. The Remote PED Client on the SafeNet Network HSM appliance or HSM host computer, receiving that message, would gracefully close the link and the host-side timeout would not be reached.

Generally, the state that causes the HSM to look for PED authentication via the Remote path, rather than from a locally-connected PED, persists unless you change it. The session between the Remote PED and the PedServer on that host also remains intact. It is the link between PedClient (at the HSM end) and PedServer (at the Remote PED end) that goes down for lack of use, and that link can be restarted with a single command when needed.

If it has been some time (more than half an hour) since you performed any authentication operations via Remote PED, the link has probably lapsed. Find out with `lunacm:>ped show`. If it says "not assigned", then the connection has been lost. Simply issue the `ped connect` command again, when needed.

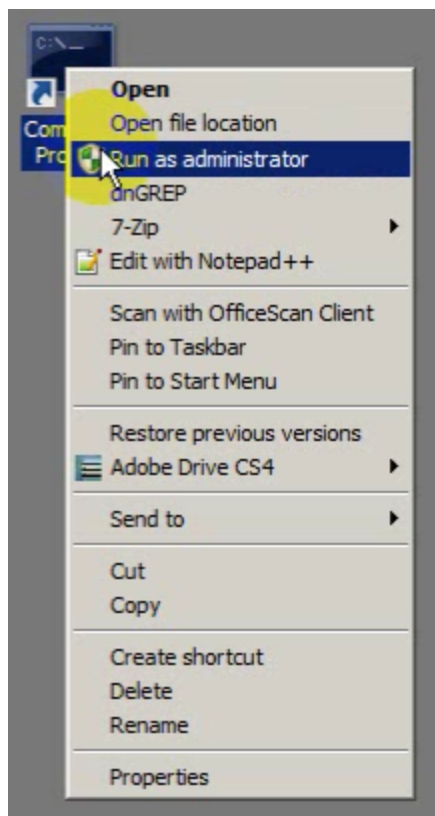
Ports

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 SafeNet 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 SafeNet 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 who has what access by invoking MofN when you first create an RPV.

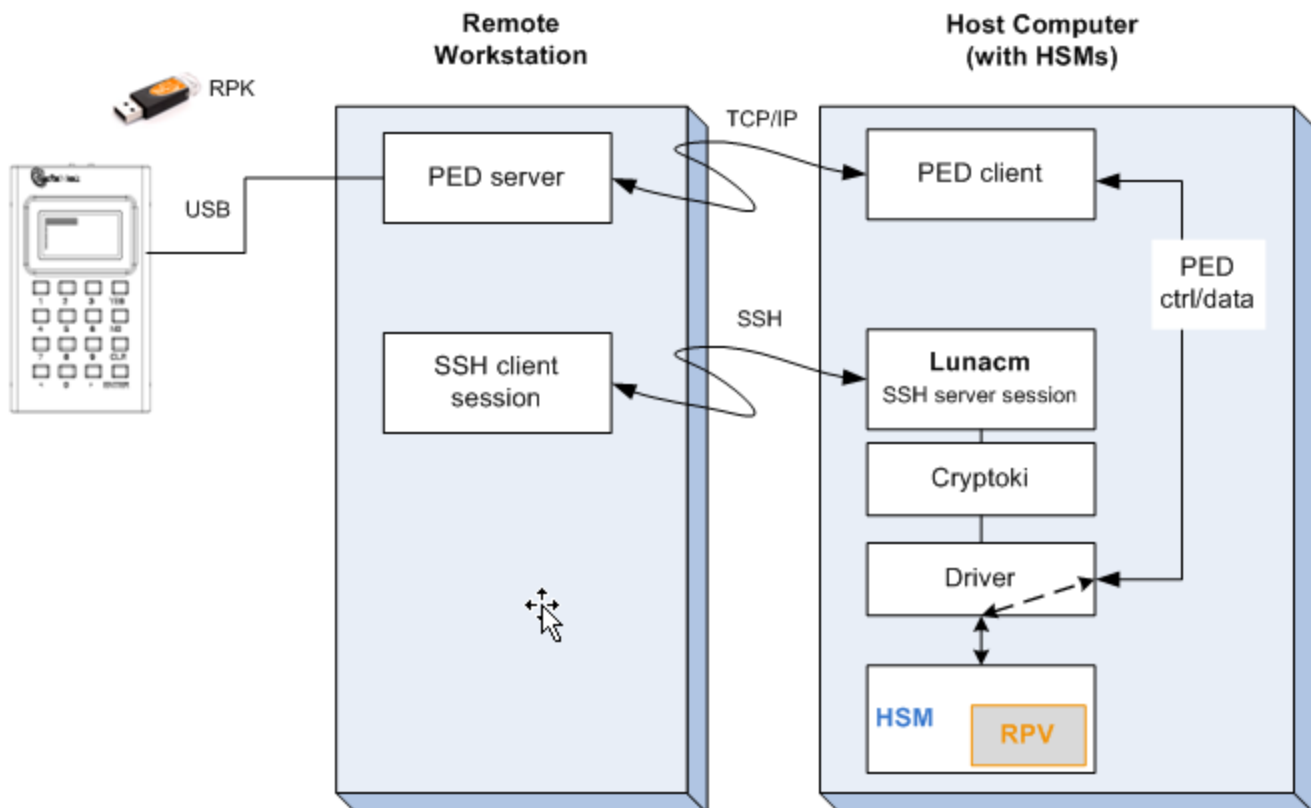
Compatibility

Remote PED for SafeNet HSM 5.2 is not compatible with earlier HSM versions.

Security of Remote PED

All communication between the Remote PED and the HSM in its host is transmitted within an AES-256 encrypted channel using session keys based on secrets (the Remote PED Vector (RPV) on the orange Remote PED Key (RPK)) that are shared out-of-band via the Remote PED role. This is considered a very secure query/response mechanism.

Remote PED Architecture



The PED client and server are software components that allow the HSM and PED to communicate via a TCP/IP network.

The PED server resides on the host computer where a remote-capable SafeNet PED is USB connected. The PED server acts as an intermediary, accepting requests and serving PED prompts and actions and data to requesting HSMs (usually located at a distance from the Remote PED and its associated PED server).

The PED Client resides on the system hosting the HSM. That could be a workstation or server with a SafeNet USB HSM connected or a SafeNet PCI-E HSM embedded, or it could be a SafeNet Network HSM appliance, any of which can request PED services from the PED Server through the network connection

Once the data path is established and the PED and HSM are communicating, they establish a common Data Encryption Key (DEK) which is used for PED protocol data encryption and authenticate each other as described below.

An authentication failure disconnects the parties.

DEK establishment is based on the Diffie-Hellman key establishment algorithm and an RPV (Remote PED Vector), shared between the HSM and the PED via the orange Remote PED iKey (RPK). Once a common Diffie-Hellman value is established between the parties via the Diffie-Hellman handshake, the RPV is mixed into the value to create a 256-bit AES DEK on each side. If the PED and the HSM do not hold the same RPV, the resulting DEKs will be different between them, and the parties won't be able to talk. This property is used in providing mutual authentication of the PED and the HSM.

Mutual authentication is achieved by exchanging random nonces, encrypted using the derived data encryption key. The authentication scheme operates as follows:

HSM	–	Remote PED
Send 8 bytes random nonce, R1, encrypted using the derived encryption key.	$\{R1 \parallel \text{padding}\}_{Ke} \rightarrow$	
	$\leftarrow \{R2 \parallel R1\}_{Ke}$	Decrypt R1. Generate an 8 byte random nonce, R2. Concatenate R2 R1 and encrypt the result using the derived encryption key.
Decrypt R2 R1. Verify that received R1 value is the same as the originally generated value. Re-encrypt R2 and return it to Remote PED.	$\{\text{padding} \parallel R2\}_{Ke} \rightarrow$	Verify that received R2 value is the same as the originally generated value.

Following successful authentication, the random nonce values are used to initialize the feedback buffers needed to support AES-OFB mode encryption of the two communications streams (one for each direction).

Sensitive data in transition between a PED and an HSM is end-to-end encrypted: plaintext security-relevant data is never exposed beyond the HSM and the PED boundaries at any time. The sensitive data is also hashed, using a SHA-256 digest, to protect its integrity during transmission.

Remote PED and pedclient and pedserver

When it is not convenient to be physically near the host computer that contains a SafeNet HSM, you can remotely and securely connect a SafeNet PED and present PED Keys, as follows:

1. On the computer used to host the HSM or [[[Undefined variable Default.SO_Partiton]]], allow remote desktop access or ssh, and have the `pedclient.exe` program available.
2. On the remote administrative workstation used to host the remote PED, (which for this purpose must run the Windows operating system) use remote-desktop client or use ssh, have a SafeNet PED2 (with Remote capability) connected, and have the `pedserver` tool installed and running.
3. Using remote desktop or ssh, make the Remote PED connection between the HSM host and the remote administrative workstation:
 - a. Start the `pedserver` listening on the remote PED host.
 - b. Start `pedclient` on the HSM host 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.

4. Run commands on the HSM 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

Remote PED (via `pedclient.exe`) can provide PED services to only one HSM slot at a time. To provide PED interaction (remotely) to another slot, you must close `pedclient.exe` for that first slot/HSM and then open `pedclient.exe` for the next slot/HSM.

Once 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 slot), 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 slot, 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

The PED is an accessory device that allows compatible SafeNet 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 SafeNet PED and PED Key actions can be accomplished with the SafeNet PED directly connected to the SafeNet 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 SafeNet PED with Remote capability.

Remote PED is supported (and requires installation/configuration) in two parts:

- PEDClient, which runs on the HSM host and allows the HSM to seek PED Key data from a remotely located SafeNet PED. PEDClient is part of the SafeNet HSM Client software installation for every type of SafeNet HSM except SafeNet Network HSM (because PEDClient is already present within the SafeNet Network HSM appliance).
- PEDServer, which runs on Remote PED host. PEDServer is installed if the "Remote PED" option is selected during SafeNet Client software installation, and includes the PedServer.exe executable, along with the SafeNet PED device drivers. If the target computer is intended to be a PEDServer, but is not going to be a Client to your SafeNet HSM, then you do not need any of the other SafeNet HSM Client software; you can use SafeNet HSM Client installer to install only the Remote PED option.

You will need:

- An HSM host, configured as described elsewhere in this document, with PEDClient available, and with its own working network connection.
- A remote PED host computer with a supported operating system (see the Customer Release Notes for supported platforms) to run PEDServer.
- Sufficient privileges on the remote PED host, depending on platform and location (local network, WAN, VPN...)
- Current SafeNet HSM Client installer (LunaClient.msi)
- SafeNet PED (Remote capable) V.2.4.0-3 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.

Configuring the PEDClient and PEDServer

This configuration takes place in two locations:

- on the HSM host.
- on the Remote PED host.

To configure the HSM host computer

1. Install/configure your HSM host as described previously.
2. Change to the directory where SafeNet HSM Client is installed and launch lunacm.

Type: `c:\Program Files\SafeNet\LunaClient> lunacm`

3. With a SafeNet PED connected locally, initialize a Remote PED Vector for the HSM and for an orange PED Key.

Type: `lunacm:> ped vector init` and respond to the SafeNet PED prompts.

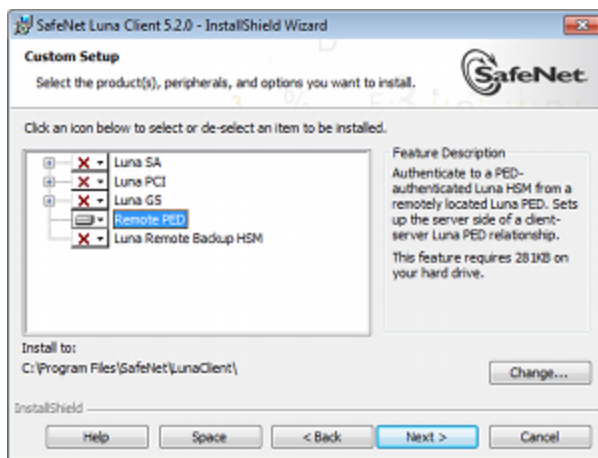
By means of your responses to the PED prompts, you can choose to have the HSM generate a new RPV to be held by both the HSM and a new orange PED Key, or you can choose to re-use an RPV already on an existing orange PED Key, and imprint that on the HSM.

As always, we suggest that you make at least one extra copy of the Remote PED Key.

4. Bring an orange PED Key, containing the RPV for this HSM, from the HSM to the location of the Remote PED server.

To configure the Remote PED host computer

1. SafeNet PED should not yet be connected to the PEDServer computer.
2. Install the SafeNet HSM Client software, selecting "Remote PED" option - for the purposes of Remote PED. Any additional SafeNet HSM Client installation choices are optional for this host system.

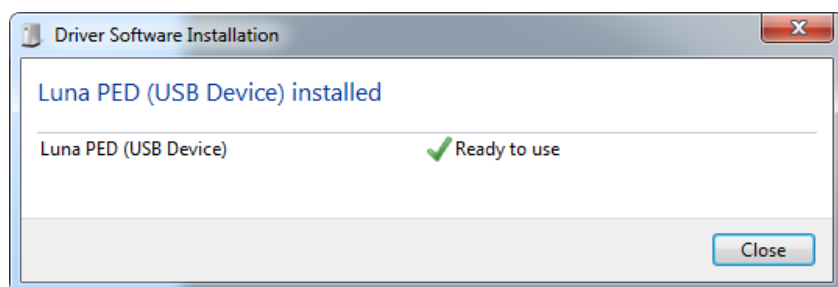


3. Click **Install** when prompted to install the driver.



4. Reboot the computer to ensure that the LunaPED driver is accepted by the operating system. This is not required for Windows Server Series.
5. Connect the Remote Capable SafeNet 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.



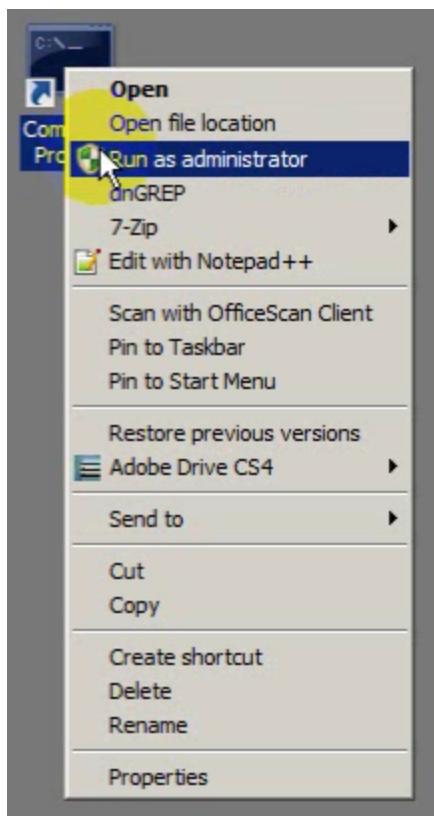
SafeNet PED performs its start-up sequence, and settles into Local Mode, by default.

6. Press the [<] key on the PED to access the "Select Mode" menu.
7. Press [7] to select "Remote PED" mode.
8. Ensure that your 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.
9. 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.

10. Go to the installed SafeNet HSM Client directory.

Type `cd "\Program Files\SafeNet\LunaClient"`

11. Launch the PEDServer.

Type `pedserver -mode start`

12. 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). In addition, "Ped2 Connection Status:" should say "Connected". This indicates that the SafeNet PED that you connected (above) was found by PEDServer.



Note: If a port other than the default 1503 was specified in `pedserver -mode start`, for example `pedserver -mode start -port 1523`, then `pedserver -mode show` command should pass in the same port, for example `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.

13. 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 SafeNet Remote PED connection at intervals of hours or days, providing the newly assigned IP address of your PEDServer computer each time.



Note: We generally advise not specifying the IP address when starting the PED server, unless you have a specific reason to set an address there. Just say "pedserver -mode start".

In a volatile network or VPN situation, this means that, when the host IP changes on the PED server, only pedclient needs restarting with the new pedserver IP address. There is no need to also stop-and-restart pedserver.exe with a new IP.

Once started, pedserver.exe remains on, and listening until you explicitly tell it to stop, or until the host computer stops.

To configure the HSM to use the remote PED



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.

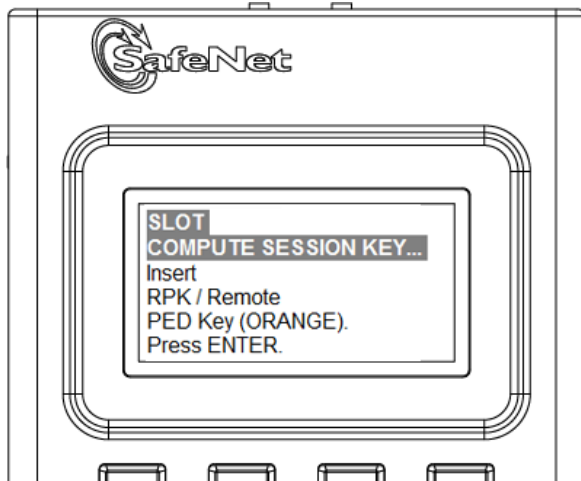
1. Launch the PEDClient on your HSM server, identifying the PEDServer instance (configured above) to which the HSM is to connect for its authentication requirements.

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`

SafeNet PED operation required to connect to Remote PED - use orange PED Key(s).

At this point, the remote SafeNet PED should come to life, briefly saying "Token found..." followed by this prompt:



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

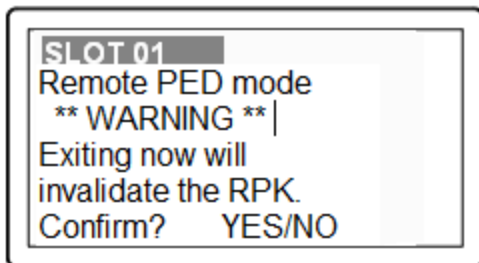
The PEDServer utility continues to run until explicitly stopped.

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:



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.

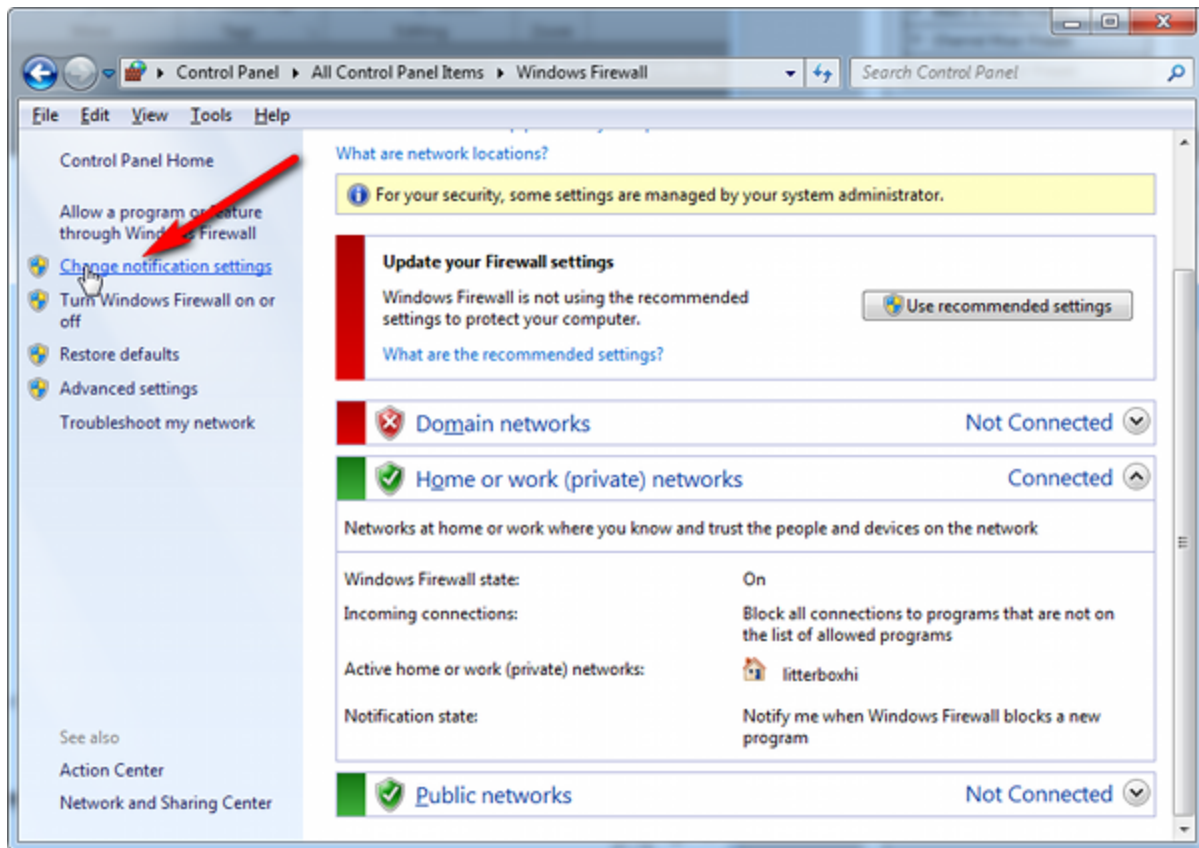
Troubleshooting

Here are some suggestions for addressing some possible issues when configuring SafeNet Remote PED.

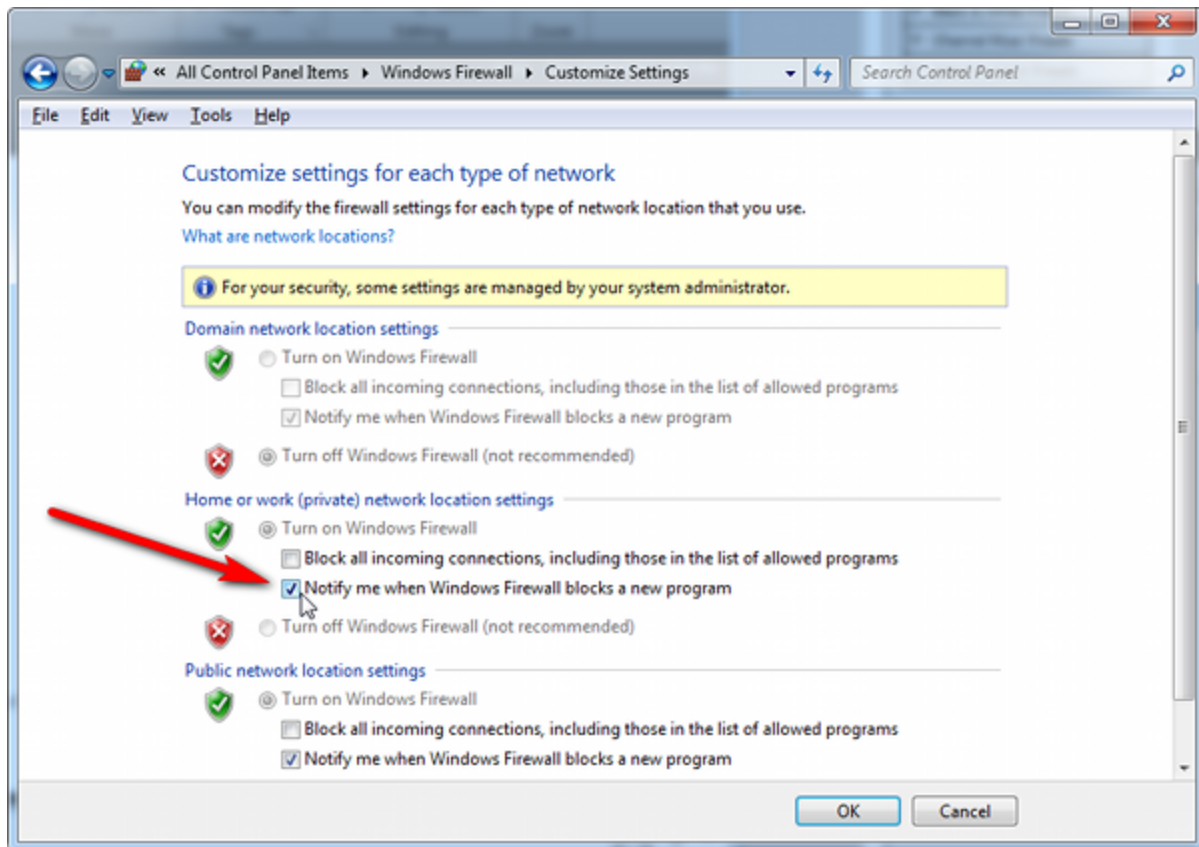
Firewall blocking

If you experience problems while attempting to configure a SafeNet Remote PED session over VPN, you might need to adjust Windows Firewall settings.

1. From the Windows Start Menu, select "Control Panel".
2. From the "Control Panel", select "Windows Firewall".
3. From the "Windows Firewall" dialog, select "Change notification settings".



4. In the dialog "Customize settings for each type of network", go to the appropriate section and activate "Notify me when Windows Firewall blocks a new program".



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 SafeNet 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.

"Jump" Server [option]

An option that some customers use is a port-forwarding "jump" server, co-located with the SafeNet HSM appliances, on the datacenter side of the firewall. The datacenter is usually a very stable/static network environment. By contrast, a client host on a desktop in a corporate office is more likely to be separated from the internet by an assortment of switches, firewalls, routers, etc., subject to change for any number of reasons. Implementing a jump server can be a low-cost and useful addition:

- to get around port-blocking problems, or to be able to react quickly to shifts in the corporate port and routing environment,

- as a way to implement a PKI authentication layer for Remote PED, and optionally for other SSH access, by (for example) setting up smart-card access control to the jump server.

For our own test of the solution, we used a standard Ubuntu Server distribution, with OpenSSH installed. No other changes were made to the system from the standard installation.

1. Connect a SafeNet PED, set to Remote Mode, to a Windows host with SafeNet HSM Client installed, and PEDServer running (see above for details).
2. From the Windows host, in an Administrator Command Prompt, run this command:

```
plink -ssh -N -T -R 1600:localhost:1503 <user>@<IP of Linux Server>
```

3. From the SafeNet Network HSM, run the command:

```
hsm ped connect -ip <IP of Linux Server> -port 1600
```

The connection is made to the Windows host running PEDServer, via the Linux Server, through the SSH session that was initiated out-bound from the Windows host.

A variant of this arrangement has port 22 also routed through the jump server, which allows you to bring administrative access to the SafeNet appliance under the PKI access-control scheme.

Using the Remote PED Feature

To use Remote PED for the first time, you will need:

- a SafeNet PED 2.4.0-3 (or later) with Remote PED feature installed (new Remote PED units are shipped with this



PED remote

sticker on the front)

- a power adapter for the Remote PED (when the PED is not connected to a SafeNet Network HSM, 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 SafeNet HSM (for the first session only)
- HSM 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

You will need physical access to your SafeNet Network HSM 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).



Note: If you encounter timeout problems (possible if you are using MofN 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.



Note: 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:>
```

Note: If the HSM host (a SafeNet Network HSM appliance or a host computer with SafeNet PCI-E HSM or SafeNet USB HSM) has more than one SafeNet HSM connected, then you might need to specify the `"-serial"` option, to identify the desired HSM by its serial number.



If `"-serial"` is not specified in commands

```
hsm ped vector init
hsm ped vector erase
hsm ped connect
hsm ped disconnect
```

then the action defaults to the first HSM that is found.

Setup Instructions

The steps to set up Remote PED are:

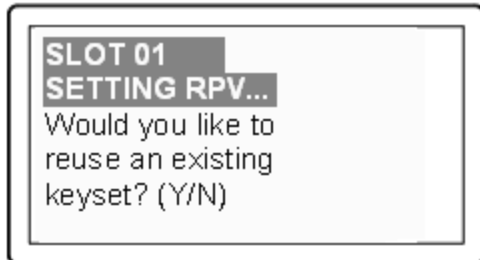
1. **Initialize the HSM** [if you have not already done so]- the creation of the orange Remote PED key requires HSM login; HSM login requires an initialized HSM, all of which must be done with a local PED connection the first time.
2. Have the SafeNet PED connected to the PED port of the HSM, and set to Local PED mode.
3. **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:>
```
4. **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... SafeNet PED operation required to initialize remote PED key vector - use orange PED key(s).
```

(At this time, go to the SafeNet 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:



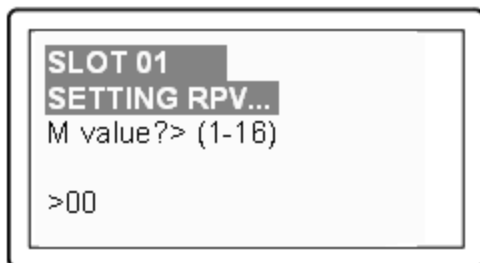
SLOT 01
SETTING RPV...
Would you like to
reuse an existing
keyset? (Y/N)

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.

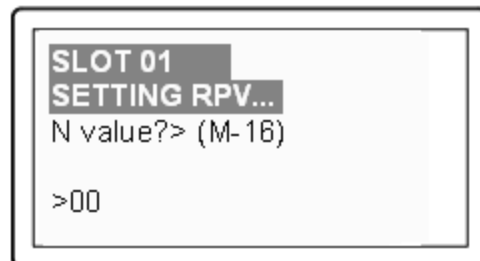
The PED says:



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].

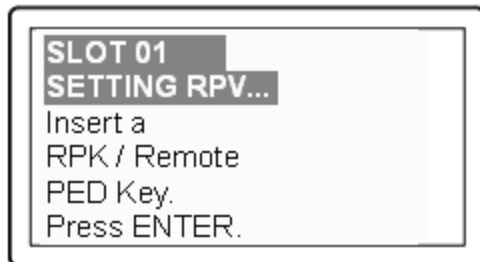


SLOT 01
SETTING RPV...
N value?> (M-16)

>00

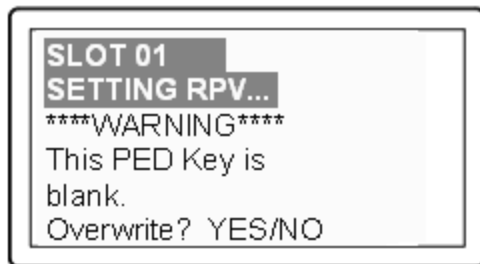
If you have invoked MofN with an M value greater than "1", then you must enter a value for N that is equal to, or

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.

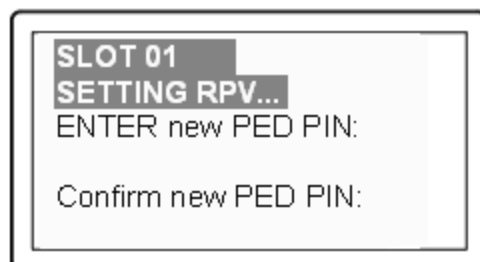


Do as prompted, inserting an unused PED Key into the PED's key slot (top-right of the PED), and press [ENTER].

For a fresh, new, never-before imprinted PED Key, the PED says:



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 MofN, 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.



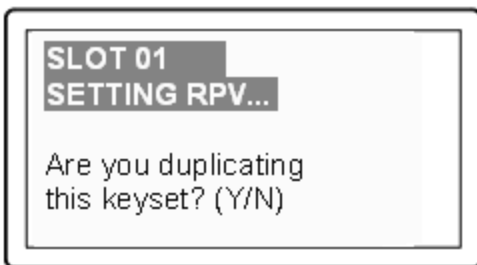
If you need two-part security to protect the Remote PED function, and wish to add a "something you know" component to the "something you have" (physical PED Key), type a series of digits on the keypad, then type them again to confirm. Now, whenever you are prompted to present the orange RPK, you must 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, SafeNet 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, SafeNet 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 shell [lunash] exchanging messages with the Remote PED Client application (also found on your SafeNet 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.)

5. 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 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 SafeNet Network HSM 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 SafeNet HSM Client installation to include Remote PED at that time.

When you connect your SafeNet 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.]).

6. Connect the Remote PED to its power source via the power adapter.
7. Connect the Remote PED to the workstation computer via the USB cable.
8. 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.
9. 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.

10. Open an ssh session to the SafeNet Network HSM appliance and login as admin.

11. 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:>
```

NOTE: the serial number option on command hsm ped connect is needed if you are using Remote PED with an HSM other than the onboard SafeNet Network HSM (such as a connected SafeNet USB HSM for PKI). If a serial number is not specified, the internal HSM is assumed by default.

12. 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:>
```

13. 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 SafeNet Network HSM/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 SafeNet Network HSM/appliance, and without having a SafeNet Network HSM PED directly attached to the SafeNet Network HSM/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 [HSM products that include Remote PED are now routinely submitted to approving agencies (like NIST/FIPS) for validation].

14. 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:>
```

Note: 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.)



Note: 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.

******* The default timeout for a Remote PED link between PedClient at the HSM and PedServer at the Remote PED, is 1800 seconds, or 30 minutes. If no Remote PED activity is requested for the entire timeout duration, the link ends, and must be re-established. While that link is down, and the HSM remains set to expect Remote PED operation, any requested PED operations simply fail. We recommend performing a disconnect before performing a connect, to ensure that the old link is cleanly severed and that a new link is cleanly established. *******



Note: 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.)

If you encounter problems with Remote PED, "Troubleshooting Remote PED" below.

Troubleshooting Remote PED

This section describes how to recognize and fix some problems that you could encounter.

Ped connect can fail if IP is not accessible

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%)

To resolve this problem

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.

VPN

If pedserver is running on a laptop that might change location, it can happen that the active network address changes, even though the laptop is not shutdown. An example might be if you unplugged from working at home, over the corporate VPN, commuted to the office, and reconnected the laptop there. The IP address that your laptop is assigned when joining directly to the office network would be different from the address that had been assigned by the VPN server.

In that situation, pedserver is still configured with the address you had while using the VPN. That pedserver session cannot work with the new address that is now assigned to your laptop. Running `pedserver -mode stop` does not completely clear all settings, so running `pedserver -mode start` again fails with a message like "Startup failed. : 0x0000303 RC_OPERATION_TIMED_OUT".

To resolve this problem

To resolve this problem,

1. Close the current Command Prompt window.
2. Open a new Command Prompt.
3. Verify the current IP address with command **ipconfig**
4. Run **pedserver -mode start -ip <new-ip-address> -port <port-number>** and it should now succeed.

Timeout

The default timeout for a Remote PED link between PedClient at the HSM and PedServer at the Remote PED, is 1800 seconds, or 30 minutes. If no Remote PED activity is requested for the entire timeout duration, the link ends, and must be re-established. While that link is down, and the HSM remains set to expect Remote PED operation, any requested PED operations simply fail. We recommend performing a disconnect before performing a connect, to ensure that the old link is cleanly severed and that a new link is cleanly established.

Pedserver fails to start with "LOGGER_init failed"

The `pedserver.exe` process must be run using administrator privileges. If you forget and accidentally launch `pedserver.exe` in an ordinary, non-privileged-user command-prompt window, the pedserver fails to work, but the process is launched and continues in the background. If you then attempt to launch `pedserver.exe` from an Administrator command prompt, it fails with message "LOGGER_init failed". The logger is a necessary service for pedserver, and has failed to initialize for the new attempt because the earlier, non-functional instance of pedserver has locked the logger.

To resolve this problem

If pedserver fails with the message "LOGGER_init failed", proceed as follows:

1. Check that the pedserver process is not already running.
2. If it is, stop the process (to free the logger service).
3. Start the pedserver process again, as Administrator.

Removing/Destroying Content for Safe Disposal

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

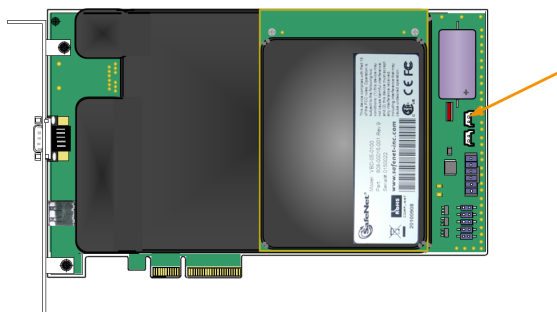
- placing the unit into storage, perhaps as a spare
- shipping to another location or business unit in your organization
- shipping the unit back to SafeNet/Gemalto for repair/re-manufacture
- removing the HSM permanently from operational use, for disposal at end-of-life

This chapter describes the available options in the following sections:

- "Declassify/Decommission the HSM Card" below
- "Resetting to Factory Condition" on the next page
- "End of service and disposal" on the next page
- "Comparison of Destruction/Denial Actions" on page 400
- "RMA and Shipping Back to SafeNet" on page 404
- "What Does Zeroized Mean?" on page 404

Declassify/Decommission the HSM Card

To Declassify/Decommission the SafeNet PCI-E HSM for disposal, or to ship back to Gemalto for repair, use the blade of a screwdriver, or other conductive tool to short-circuit the two pins of the Decommission header (J3) on the HSM card, located near the battery.



Resetting to Factory Condition

The command `hsm factoryReset` (which can be run from a local serial console only) affects only the HSM, and not the settings for other components of the appliance.

The command `(sysconf config factoryReset)` affects appliance settings external to the HSM.

To bring your entire SafeNet Network HSM as close as possible to original configuration, as shipped from the factory, run both commands.

"Original factory configuration" is an approximation. If you have performed firmware and software updates, those remain in place, and are not affected by the commands described above. The reset commands (above) affect contents of the HSM and settings of the HSM and the appliance that contains it. Reverting of software and firmware is outside the scope.

End of service and disposal

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

Needs Can Differ

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

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

SafeNet HSM Protects Your Keys and Objects

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

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

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

How the HSM encryption keys protect your sensitive objects

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

If the HSM experiences a Decommission event (pressing of the small red button on back of SafeNet Network HSM, or shorting of the pins of the decommission header on the K6 HSM card, or removal of the battery while main power is not connected to a SafeNet USB HSM) then the KEK is deleted.

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

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

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

Comparison of Destruction/Denial Actions

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

Event	MTK is destroyed HSM is unavailable, but use/access can be recovered after reboot (See Note 1)	KEK is destroyed (Real-Time Clock and NVRAM) HSM contents cannot be recovered without restore from backup (See Note 2)	Reset appliance admin password	How to discover (See Note 3)	How to recover
- three bad SO login attempts or - lunash:> hsm zeroize (for SafeNet Network HSM), lunacm:> hsm zeroize (for SafeNet PCI-E HSM and SafeNet USB HSM) or - lunash:> hsm factoryReset (for SafeNet Network HSM),	NO	YES	NO	hsm.log entry or "Zeroized: Yes" in HSM Information (from hsm show command)	Restore HSM objects from Backup

Event	MTK is destroyed HSM is unavailable, but use/access can be recovered after reboot (See Note 1)	KEK is destroyed (Real-Time Clock and NVRAM) HSM contents cannot be recovered without restore from backup (See Note 2)	Reset appliance admin password	How to discover (See Note 3)	How to recover
lunacm:> hsm factoryreset (for SafeNet PCI-E HSM and SafeNet USB HSM) or - any change to a destructive policy or - firmware rollback (See Note 4)					
hardware tamper - undervoltage or overvoltage during operation or - under-temperature or over-temperature during operation or - chassis interference (such as cover, fans, etc.) OR software (command- initiated) tamper - lunash:> hsm srk transportMode enter or - lunacm:> srk transport	YES	NO	NO	Best practice - have external MTK split on SRK (purple PED Key), which forces administrative intervention to recover from tamper. Otherwise, parse hsm.log for text like "tamper", "TVK was corrupted", or "Generating new TVK", indicating that a tamper event was logged. Example: RTC: external tamper latched/MTK: security function was zeroized on previous tamper event and has not been	Reboot [See Note 1]

Event	MTK is destroyed HSM is unavailable, but use/access can be recovered after reboot (See Note 1)	KEK is destroyed (Real-Time Clock and NVRAM) HSM contents cannot be recovered without restore from backup (See Note 2)	Reset appliance admin password	How to discover (See Note 3)	How to recover
				restored yet ... also, keywords in hsm.log like: "HSM internal error", "device error" SafeNet Network HSM appliance front panel flashes error 30.	
decommission (shorting-circuiting the tamper2 header pins on SafeNet PCI-E HSM, or pressing the Decommission button on the back of the SafeNet Network HSM appliance, which shorts tamper2, or unplugging main power and removing the battery from SafeNet USB HSM)	NO	YES	NO	Look for log entry like: RTC: tamper 2 signal/Zeroizing HSM after decommission...LOG (INFO): POWER-UP LOG DUMP END	Restore HSM objects from Backup
<p>Note 1a: MTK is an independent layer of encryption on HSM contents, to manage tamper and Secure Transport Mode. A destroyed MTK is recovered on next reboot if no external split (SRK) of the recovery vector exists (that is, if both portions of the recovery key are available inside the HSM).</p> <p>A destroyed MTK can be recovered if one of the recovery components has been moved outside the HSM onto a secure recovery key (SRK), and that SRK can be presented via SafeNet PED at the next HSM reboot.</p> <p>If MTK cannot be recovered, only restoring from backup onto a new or re-manufactured HSM can retrieve your keys and HSM data.</p>					

Event	MTK is destroyed HSM is unavailable, but use/access can be recovered after reboot (See Note 1)	KEK is destroyed (Real-Time Clock and NVRAM) HSM contents cannot be recovered without restore from backup (See Note 2)	Reset appliance admin password	How to discover (See Note 3)	How to recover
<p>Note 1b: If an external SRV exists, and you wish to stop using it (having no need for Secure Transport Mode, and no need for enforced administrative response in case of a tamper event), you must bring the external SRV back into the HSM with srk disable command. You may not simply destroy or overwrite the purple PED Key without first ensuring that the HSM is in possession of all the MTK recovery components.</p> <p>Note 2: KEK is an HSM-wide encryption layer that encrypts all HSM objects, excluding only MTK, RPK, a wrapping key, and a couple of keys used for legacy support. A destroyed KEK cannot be recovered. If the KEK is destroyed, only restoring from backup can retrieve your keys and HSM data.</p> <p>Note 3: To check the health of a remote HSM, script a frequent login to the HSM host and execution of a subset of HSM commands. If a command fails, check the logs for an indication of the cause.</p> <p>Note 4: These actions all create a situation where hsm init is required, or strongly recommended before the HSM is used again.</p>					

In addition, another event/action that has a destructive component is HSM initialization, which can be of either the "soft" or "hard" variety.

- HSM init is soft if you have not performed an **hsm factoryReset** before **hsm init**.
- HSM init is hard if it is performed following **hsm factoryReset**.

Either way, HSM and partition objects are gone, so only a restore from backup can bring them back. Effects of soft versus hard initializations are summarized below :

Condition/Effect	Soft init	Hard init
SO authentication required?	Yes	No
Can set new HSM label	Yes	Yes
Creates new SO identity	No	Yes

Condition/Effect	Soft init	Hard init
Creates new Domain	No	Yes
Destroys partitions	Yes	No (none exist to destroy) *
Destroys SO objects	Yes	No (none exist to destroy) *

* hsm factoryReset was performed, and destroyed partitions and objects, before the hard init... otherwise, it could not be a hard init.

RMA and Shipping Back to SafeNet

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

You would deal through your SafeNet representative to obtain the Return Material Authorization (RMA) and instructions for packing and shipping.

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

Two actions that you can take are:

- Press the "decommission" button on the appliance back panel; this forcibly clears all HSM contents.
- If the appliance uses PED (Trusted Path) authentication, set the HSM into Secure Transport Mode (`hsm srk enable` (if not already enabled) followed by `hsm srk transportMode enter`), and simply do not send us the purple key. We have no way to access the HSM, and no choice but to remanufacture it.

What Does Zeroized Mean?

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

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

Therefore,

- if you perform `hsm factoryReset`, or
- if you make too many bad login attempts on the SO account, or
- if you short the pins of the decommission header (indicated by the orange arrow in the picture), or
- if you set a "destructive" HSM policy, or
- if you perform HSM firmware rollback,

not only are any temporarily decrypted keys destroyed, but all customer keys on the HSM are immediately rendered inaccessible and unrecoverable.

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



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

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

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

- the Master Tamper Key, which enables tamper handling as well as Secure Transport Mode,
- the Remote PED Vector, to allow Remote PED-mediated recovery from tamper or from Secure Transport Mode,
- the hardware origin key that certifies the HSM hardware as having been built by Gemalto-SafeNet, and
- a couple of legacy keys that allow objects to be migrated from legacy HSMs.

User and Password Administration

This section describes tasks related to identities in the HSM or HSM partitions, including changing and resetting passwords, events or actions that cause HSM contents to be lost, and so on. It contains the following sections:

- "About Changing HSM and Partition Passwords" below
- "Failed Logins" on page 408
- "Resetting Passwords" on page 410
- "Default Challenge Password" on page 413

About Changing HSM and Partition Passwords

From time to time, you might have reason to change the various passwords on the appliance and HSM. This might be because a password has possibly been compromised, lost, or forgotten, or it might be because you have security procedures that mandate password-change intervals.

The two options are:

Action	Description	When used
Resetting PW	A higher authority sets a user's credentials back to a known default value (without requiring the knowledge or cooperation of the affected user),	<ul style="list-style-type: none"> • current holder has lost or forgotten his/her credential (forgot a password, misplaced a PED Key) • current credential is known or suspected to have become compromised • current holder has departed organization
contrasts with...		
Changing PW	The legitimate holder of the credential is able to log in with current credentials before directing the HSM, under the current logged-in user's own authority, to change that user's credential to a new value.	<ul style="list-style-type: none"> • credential holder suspects possible compromise of credential • credential holder is complying with organization security provisions (such as mandatory password-change interval)

HSM Passwords

Resetting HSM Password

There is no provision to reset the HSM Admin password (for Password Authentication) or PED Key (for Trusted Path), except to re-initialize the HSM, which zeroizes the contents of the HSM and of all Partitions on that HSM.

Resetting the password/authentication of a role or user requires a higher authority to invoke the reset. On the HSM, there is no authority higher than the SO / HSM Admin.

Changing HSM Password

To change the HSM password (for Password Authentication) or the secret on the blue PED Key (for Trusted Path), you must log in as HSM Admin using the current password (or blue PED Key). This is prompted by the `hsm changePw` command, so you do not need to log in separately.

```
lunash:> hsm changePw
Luna PED operation required to login as HSM Administrator - use Security Officer (blue) PED key.
Command result : (0) success
lunash:>
```

Partition Passwords

A deliberate change to a Partition password is different from a [password reset](#). In both cases, the Partition or HSM contents remain intact.

Resetting Partition Password

- you must be logged in as HSM Admin, 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).

```
lunash:> partition resetPw -newpw mynewpw -partition mypartition1
```

Changing Partition Password

- you do not have to be logged in as HSM Admin or SO, but
- you do need to know the current Partition password. For Trusted Path HSMs, you must provide the current black PED Key.

```
lunash:> partition changePw -newpw mynewpw -oldpw myoldpw -partition mypartition1
```

You can choose not to include the passwords with the command, which:

- causes the system to prompt for old and new passwords (obscuring them with asterisks (*) for greater security, and
- presents additional options as shown in the example below.

For a PED-authenticated HSM, the following example changes only the challenge secret of the named partition, and leaves the black PED Key contents unchanged.

Example:

```
[myluna] 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:

> *****

Please re-enter password to confirm:

> *****

Luna PED operation required to activate partition on HSM - use User or Partition Owner (black) PED key.

'partition changePw' successful.

Command Result : 0 (Success)

[myluna] lunash:>

Failed Logins and Forgotten Passwords

"Failed Logins" below.

Appliance

For password changes affecting the appliance, not including the HSM .

Failed Logins

If you fail three consecutive login attempts as HSM Security Officer (or SO), the HSM contents are rendered unrecoverable. This is a security feature (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.

Note: The system must actually receive some erroneous/false information before it logs a failed attempt – if you merely forget to insert a PED Key (for PED-authenticated HSMs), or inserted the wrong color key, that is not counted as a failed attempt.



To fail a login attempt on a Password-authenticated HSM, you would need to type an incorrect password. To fail a login attempt on a PED-authenticated HSM, you would need to insert an incorrect PED Key of the correct color, or to type an incorrect PED PIN, if one had been set for that PED Key.

As soon as you successfully authenticate, the counter is reset to zero.

View a table that compares and contrasts various "deny access" events or actions that are sometimes confused. See "Comparison of Destruction/Denial Actions" on page 400.

Other roles and functions that need authentication on the HSM have their own responses to too many bad authentication attempts. Some functions do not keep a count of bad attempts; the simple failure of a multi-step or time-consuming operation is considered sufficient deterrent to a brute-force attack. The table in the next section summarizes the responses.

HSM Response When You Reach the Bad-attempt Threshold

Role	Threshold (number of tries)	Result of too many bad login attempts	Recovery
HSM SO	3	HSM is zeroized (all HSM objects identities, and all partitions are gone)	HSM must be reinitialized. Contents can be restored from backup(s).
Partition SO	3	Partition is zeroized.	Partition must be reinitialized. Contents can be restored from backup.
Audit	10	Lockout	Unlocked automatically after 10 minutes.
Crypto Officer [Note 1]	10 (can be decreased by SO,)	Lockout	Must be unlocked/reset by the partition's SO.
Crypto User [Note 2]	10 (can be decreased by SO)	Lockout	Must be unlocked/reset by the partition's CO.
Domain	n/a	Operation fails	Retry the operation with the correct Domain - usually that would be a backup or restore
Remote PED Key	n/a	Operation fails	Retry establishing a Remote PED connection, providing the correct orange PED Key (PED-authenticated only).
Secure Recovery Key	n/a	Recovery from tamper or Secure Transport Mode fails. Entire HSM is locked. The only operation that is not locked out is establishing a Remote PED connection.	Retry recovery from tamper or from STM, providing the correct purple PED Key (PED-authenticated only).

[Note 1] If the policy "SO can reset PIN" is on, then this user is locked. If "SO can reset PIN" is off, then this user is deleted - as is any user that depends upon it, specifically the Crypto User.

[Note 2] The Crypto User is created by the Crypto Officer. Therefore, only the Crypto Officer, and not the SO of the partition, is able to reset the Crypto User. If the policy "SO can reset PIN" is off, then this user is deleted, rather than

Role	Threshold (number of tries)	Result of too many bad login attempts	Recovery
------	-----------------------------	---------------------------------------	----------

locked out when too many bad attempts are made on the CU. Similarly, if too many bad attempts are made on his creator the Crypto Officer, and that role is deleted, then the associated CU is also deleted.

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 become inaccessible and the partition must be recreated.

“Ignore failed challenge responses” can be set per partition, which ensures that failed HSM Partition Password attempts do not cause the “failed login attempt” counter to increment.

Resetting Passwords

Resetting is normally done by a higher power when an authentication secret is lost/forgotten, or compromised, and is discussed separately from merely changing authentication when the user is in legitimate possession of the current authentication.

HSM

There is no provision to reset the SO or HSM Admin password (for Password Authenticated HSMs) or the blue PED Key (for PED Authenticated or Trusted Path HSMs), 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 `hsm changePw` command, but that requires that you know the current password (or have the current blue PED Key).

The assumption, from a security standpoint, is that if you no longer have the ability to authenticate to the HSM (because you forgot the password or lost the PED Key, or because an unauthorized person has changed the password or PED Key), then the HSM is effectively compromised and must be re-initialized. Thus, no explicit “reset” command is provided.

The `hsm init` command does not require a login, and the `hsm login` command is not accepted if the HSM is in zeroized state.

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 login 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 login 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 Authentication (uses SafeNet PED and PED Keys):

One bad login

With or without `-force` (no difference):

Caution: You have only TWO HSM Admin login 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:

```
[myLuna] lunash:>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.

Default Challenge Password

Ordinarily, when a PED-authenticated HSM partition is created, a random challenge-secret is generated and displayed by the PED. That secret becomes the authentication secret for your applications accessing that HSM partition. This is a security feature and is useful for that reason in most situations. It enforces a high-security password at the outset. However, in some scenarios, the imposed hands-on activity of reading a 16-character string from the PED screen and recording it, by hand-writing or typing, might be inappropriate.

PED- authenticated SafeNet HSMs 5.4.x and later allow you to specify a default partition password at "partition create" time.

This feature is useful in three situations:

- It allows you to deploy many partitions automatically.
- It allows fully automated testing for PED-authenticated SafeNet HSMs.
- It allows the use of Crypto Command Center (CCC) to create a High Availability group, which requires all member partitions to share the same password.

The automated testing is important to us, for repeatability and reliability of our testing at various stages of development, validation, and production quality control, but many customers might also wish to perform their own automated testing after receiving purchased SafeNet HSMs, before deploying in their own networks, or after pre-configuring HSMs and partitions for shipment/deployment to their own third-party customers.

Security Effects of Administrative Actions

Actions that you take, in the course of administering your SafeNet HSM, can have effects, including destruction, on the roles, the spaces, and the contents of your HSM and its application partition(s). It is important to be aware of such consequences before taking action.

Overt Security Actions

Some actions in the administration of the HSM, or of an application partition, are explicitly intended to adjust specific security aspects of the HSM or partition. Examples are:

- changing a password
- modifying a policy to make a password or other attribute more stringent than the original setting.

Those are discussed in their own sections.

Actions with Security- and Content-affecting Outcomes

Other administrative events have security repercussions as included effects of the primary action, which could have other intent. Some examples are:

- HSM factory reset
- HSM zeroize
- change of a destructive policy
- installation/application of a destructive Capability Update
- HSM initialization
- application partition initialization

This group of administrative actions is compared in this current section "[Summary of Outcomes of Security-affecting Actions](#)" on the next page.

Elsewhere

Certain other actions can sometimes cause collateral changes to the HSM, like firmware rollback and update. They usually do not affect contents, unless a partition is full and the action changes the size of partitions or changes the amount of space-per-partition that is taken by overhead/infrastructure, such as when going to HSM firmware 6.22.0 from an earlier version. These are discussed elsewhere.

Summary of Outcomes of Security-affecting Actions

This table lists some major administrative actions that can be performed on the HSM, and compares relevant security-related effects. Use the information in this table to help decide if your contemplated action is appropriate in current circumstances, or if additional preparation (such as backup of partition content, collection of audit data) would be prudent before continuing.

Factory Reset HSM With Firmware <6.22.0

Domain	Destroyed
HSM SO Role	Destroyed
Partition SO Role	Destroyed
Auditor Role	Destroyed
Partition Roles	Destroyed
HSM or Partition/Contents	HSM/Destroyed
HSM Policies	Unchanged
RPV	Unchanged
Messaging	You are about to factory reset the HSM. All contents of the HSM will be destroyed. HSM policies and remote PED vector left unchanged.

Factory Reset HSM With Firmware ≥6.22.0

Domain	Destroyed
HSM SO Role	Destroyed
Partition SO Role	Destroyed
Auditor Role	Destroyed
Partition Roles	Destroyed
HSM or Partition/Contents	HSM/Destroyed
HSM Policies	Reset
RPV	
Messaging	You are about to factory reset the HSM. All contents of the HSM will be destroyed. HSM policies will be reset and the remote PED vector will be erased.

Zeroize HSM With Firmware ≥6.22.0

Domain	Destroyed
HSM SO Role	Destroyed
Partition SO Role	Destroyed
Auditor Role	Unchanged
Partition Roles	Destroyed
HSM or Partition/Contents	HSM/Destroyed
HSM Policies	Unchanged
RPV	Unchanged
Messaging	You are about to zeroize the HSM. All contents of the HSM will be destroyed. HSM policies, remote PED vector and Auditor left unchanged.

Change Destructive HSM Policy

Domain	Unchanged
HSM SO Role	Unchanged
Partition SO Role	Destroyed
Auditor Role	Unchanged
Partition Roles	Destroyed
HSM or Partition/Contents	HSM/Destroyed
HSM Policies	Unchanged except for new policy
RPV	Unchanged
Messaging	You are about to change a destructive HSM policy. All partitions of the HSM will be destroyed.

Apply Destructive CUF Update

Domain	Destroyed
HSM SO Role	Destroyed
Partition SO Role	Destroyed
Auditor Role	Unchanged
Partition Roles	Destroyed

HSM or Partition/Contents	HSM/Destroyed
HSM Policies	Unchanged
RPV	Unchanged
Messaging	You are about to apply a destructive update. All contents of the HSM will be destroyed.

HSM Initialize When Admin Not Initialized

Domain	Destroyed
HSM SO Role	Destroyed
Partition SO Role	Destroyed
Auditor Role	Unchanged
Partition Roles	Destroyed
HSM or Partition/Contents	HSM/Destroyed
HSM Policies	Unchanged
RPV	Unchanged
Messaging	You are about to initialize the HSM. All contents of the HSM will be destroyed.

HSM Initialize When Admin Initialized

Domain	Unchanged
HSM SO Role	Unchanged
Partition SO Role	Destroyed
Auditor Role	Unchanged
Partition Roles	Destroyed
HSM or Partition/Contents	HSM/Destroyed
HSM Policies	Unchanged
RPV	Unchanged
Messaging	You are about to initialize the HSM that is already initialized. All partitions of the HSM will be destroyed. You are required to provide the current SO password.

Non-Admin Partition Initialize When the Partition is Not Initialized

Domain	Unchanged
HSM SO Role	Unchanged
Partition SO Role	Destroyed
Auditor Role	Unchanged
Partition Roles	Destroyed
HSM or Partition/Contents	Partition/Destroyed
HSM Policies	Unchanged
RPV	Unchanged
Messaging	You are about to initialize the partition. All contents of the partition will be destroyed.

Non-Admin Partition Initialize When the Partition is Initialized

Domain	Unchanged
HSM SO Role	Unchanged
Partition SO Role	Destroyed
Auditor Role	Unchanged
Partition Roles	Destroyed
HSM or Partition/Contents	Partition/Destroyed
HSM Policies	Unchanged
RPV	Unchanged
Messaging	You are about to initialize the partition that is already initialized. All contents of the partition will be destroyed. You are required to provide the current Partition SO password.

Secure Transport Mode

This chapter describes Secure Transport Mode, the Master Tamper Key (MTK) and Secure Recovery Key (SRK), and the purple PED key. It contains the following sections:

- "MTK and SRK" below
- "Secure Transport Mode [Local]" on page 423
- "Secure Transport Mode [Remote]" on page 426
- "Re-Split Required" on page 432
- "Interrupted SRK Re-split Operation" on page 1

MTK and SRK

Every SafeNet HSM has a Master Tamper Key (MTK) that strongly encrypts all sensitive data generated and stored within the HSM. While the master tamper key remains valid, the HSM uses it to decrypt HSM contents in order to perform cryptographic operations. The master tamper key is unique for each HSM.

When the master tamper key is created, two splits of that secret are stored in separate locations within the HSM.

A tamper event erases the HSM's master tamper key, making the HSM unusable and its contents inaccessible. That is, all contents remain encrypted and the HSM cannot use the master tamper key to decrypt them while the master tamper key does not exist.

The master tamper key can be reconstituted from the two splits for resumption of HSM operation.

The scenarios are:

- a) the two splits remain inside the HSM, and the HSM can use them to immediately recover the master key
- b) one of the splits remains inside the HSM, but the other is moved out to an external device; the HSM cannot recover the HSM's master key until the external split is re-introduced into the HSM via the PED.



Note: The MTK is NOT, in any sense, any kind of "device master key". Its purpose is to implement the tamper behavior and the related Secure Transport Mode. If someone managed to get the MTK for an HSM, they have nothing really usable. All HSM contents are further encrypted by additional layers of authentication and other strong encryption, as described elsewhere in these documents.

Tamper and Recover with Purple Key NOT Enabled

If a tamper event occurs, the event is logged and the HSM stops responding. A restart is required in order to resume. The master key is reconstituted from its component splits and re-validated, making the HSM usable again. The event is recorded in the log.

The intent is to make you aware that a tamper has occurred (the log gets an entry, and the HSM waits for a restart) but not to cause ongoing inconvenience.

That scenario applies to Password-authenticated HSMs, as well as to PED-authenticated HSMs that do not have the purple SRK option invoked - both splits of the master key remain inside the HSM.

```
[myluna] lunash:>hsm srk show
      Secure Recovery State flags:

=====
External split enabled:  no
SRK resplit required:   no
Hardware tampered:      no
Transport mode:         no
Command Result : 0 (Success)
[myluna] lunash:>
```

Tamper and Recover with Purple PED Key Enabled

Some operational environments require more rigorous response to a tamper event. Your operational and security policy might require that, in addition to logging a tamper event, the HSM must stop performing until you take authoritative action to confirm your acknowledgement and clear the tamper before operations can resume.

PED-authenticated HSMs have the option to store a split of the master key, called the Secure Recovery Vector (SRV) outside of the HSM on a physical device, the Secure Recovery Key (SRK). The situation is as above:

- master key encrypts everything,
- HSM can operate on its contents while the master key is valid,
- master key splits exist,
- master key can be reconstituted and re-validated],

But one split of the master key is imprinted onto a purple PED Key (or Keys, if you choose MofN), the SRK, and not inside the HSM. In that case, the master key component stored inside the HSM is insufficient to reconstitute the master key without the portion stored externally on the purple key(s).

```
[myluna] lunash:>hsm srk show
      Secure Recovery State flags:

=====
External split enabled:  yes
SRK resplit required:   no
Hardware tampered:      no
Transport mode:         no
Command Result : 0 (Success)
[myluna] lunash:>
```

Where the purple PED Key has been invoked, a tamper event invalidates the master key, but the HSM cannot immediately re-validate because not enough splits are available. You must invoke the LunaSH command `hsm srk transportMode recover`, and present the purple PED Key (or Keys) when prompted. The PED reads the purple key(s) and provides the missing piece (the SRV) to the HSM. The HSM combines the provided external component with the internal component and reconstitutes and validates the master key. You can resume using the HSM with no loss of crypto objects.

Behavior with Purple PED Key Enabled but MISSING or DAMAGED

If SRK is enabled (meaning a split of the MTK has been moved off of the HSM, onto a purple PED Key), and you have lost or destroyed the purple PED Key, then you **cannot** bring the HSM back from tamper or Secure Transport Mode. Your only option is to return the HSM to SafeNet for re-manufacturing.

However, you CAN run `hsm factoryReset` (from a local console connection), if your security policies require active destruction of HSM objects before it leaves your possession.

When the HSM is returned to you, after re-manufacture, it has a new MTK as if it was a new HSM, so even a copy of the old purple PED Key would be of no use to an unauthorized person. You can freshly initialize the re-manufactured HSM using your existing Domain PED Key, so that you can recover objects from one of your backups onto the re-manufactured HSM.

Secure Transport Mode

If your practice is to configure and prepare your HSMs at a central location before shipping them out to remote locations in your organization, or to configure and prepare your HSMs at a central location before shipping them out to your customers then you can invoke an enhanced security option for shipping.

By switching on Secure Transport Mode (STM), you effectively create a controlled tamper event. The HSM can be shipped in a condition that is completely unusable by anyone who might intercept it if they are not in possession of the unique SRK for that HSM. At the destination, the purple PED Key is used to reconstitute the Secure Recovery Vector, and the MTK is recovered. Your recipient begins using the HSM, with your loaded keys and certificates intact, secure in the knowledge that the HSM has not been attacked in transit.

```
[myluna] lunash:>hsm srk show
      Secure Recovery State flags:

      =====
      External split enabled:  yes
      SRK resplit required:   no
      Hardware tampered:      no
      Transport mode:         yes
Command Result : 0 (Success)
[myluna] lunash:>
```

As a service for customers who request it, SafeNet can invoke Secure Transport Mode at our factory. Your HSM is shipped in anti-tamper packaging via a trusted courier. Meanwhile, the unique purple key is shipped via a separate channel, so that the two are never available together in transit. As a further precaution, we send you the verification code via yet another path. In this way, you enjoy the maximum possible assurance that your HSM has not been attacked between our facility and yours.

Similarly, if your organization ships HSMs to your customers, you can offer them the same service, with new unique purple (SRK) keys generated at each stage, if desired.

Make a New Purple PED Key (SRK external split)

In the event that a SafeNet HSM appliance has been shipped to you, in Secure Transport Mode, and you have recovered from Transport Mode, OR you have decided to invoke Transport Mode to store or ship your HSM, you might wish to generate a unique, new SRK split onto a purple PED Key. That is, the existing MTK is split in two again (different splits than previous) and one new split replaces the internally stored component while the other new split is stored on a new purple PED Key (or keys).

Use the LunaSH command `hsm srk keys resplit` and have a different purple key ready for imprinting.

The existing SRK is needed to validate the operation, but the HSM will not overwrite that purple key with the new split. This is a safety measure, in case the operation was interrupted before completion. If that happens, the old SRK remains valid and the HSM can still be used while you begin the re-split operation again. Once the re-split is successful, with a new SRK on a new purple key, the old purple key is of no further interest because its contents have been superseded [It could be used in a future re-split operation for this HSM, or it could be imprinted from another HSM, if desired].

For example, when we (SafeNet) ship an appliance in Secure Transport Mode, we make just the one purple key that we ship to you, unless requested otherwise by you. We think that we are trustworthy, but your security auditors might be required by policy or regulation to assume otherwise. For this reason, you can perform `hsm srk keys resplit` as soon as you receive your HSM, and it would not matter if we had kept additional copies of your purple key. They would be rendered useless by the re-splitting operation on your HSM.

Similarly, your customers can do the same once they receive an HSM in Secure Transport Mode from you.

If your application requires that you ship securely, but retain control of the HSM appliances, you can use the Remote PED option to retain all PED Keys (including the purple) and use them from your location when operating the remotely located HSM.

Master Key must be present

You can perform a re-splitting operation only if the HSM's master key is currently valid. That is, for security reasons, you cannot generate a new split and new purple key while an HSM is tampered and not recovered, or while an HSM is in Secure Transport Mode and not recovered. This prevents anyone not in possession of the correct purple PED Key from circumventing the HSM "lock-out" due to tamper or to STM.

What if the purple SRK has been lost?

If the purple SRK is lost, and you do not have a backup of it, then you must obtain an RMA and return the HSM to SafeNet for re-manufacture; any objects that existed on the HSM are lost, unless you have backups. For SafeNet Network HSM, some customers might prefer the additional assurance of pressing the red factory reset button on the back of the appliance, before shipping. For SafeNet PCI-E HSM, the equivalent action is shorting/shunting the two pins of the reset header (Tamper 2) on the HSM card, near the battery.

Disabling SRK

If SRK has been enabled (one of the HSM's master key splits has been moved to an external purple PED Key), you can choose to disable the feature.

To do so, you must provide the appropriate purple SRK(s) when you run the command `hsm srk disable`, so that the external split can be read back into a secure internal location.

From that point on, the HSM behaves in the same way as a password-authenticated HSM with respect to tamper events:

- a real tamper event is merely noted in the log and does not hamper HSM operation (beyond requiring restart and re-login)
- Secure Transport Mode is not possible (until SRK is re-enabled).

Compare and Contrast Some "Denial" and Destructive Scenarios

View a table that compares and contrasts various "deny access" events or actions that are sometimes confused - "Comparison of Destruction/Denial Actions" on page 400.

Secure Transport Mode [Local]

This topic describes what to do if you wish to invoke Secure Transport Mode (STM) on a local SafeNet Network HSM, when shipping the appliance:

- to your customer or
- to your partner organization or
- to your own personnel at another site within your organization.

This page applies to PED Authenticated HSMs only. It does not apply to Password Authenticated HSMs.

As well, you could use STM for securely storing the HSM, where "transport" would take place simply into, and later out of, your warehouse or vault. However, you would also need to manage separate secure storage and handling of the imprinted purple PED Key (SRK) for that HSM until it was time to recover the HSM and return it to service.

BACKUP

Perform backups of your partitions before you continue with Secure Transport Mode procedure (see "[Backup and Restore HSMs and Partitions](#)" on page 39).

First time - no existing SRK (purple key)

This section describes the procedure if you are performing all the actions locally - that is, if your SafeNet PED is connected directly to the SafeNet Network HSM appliance when you invoke STM.

This procedure performs a new split of the HSM's master key and creates a new external split (SRV) to be imprinted on a new purple PED Key. This ensures that nobody could have a pre-existing copy. If you do wish to re-use an existing SRK, do not disable SRK - [re-]enabling SRK includes a re-split of the HSM's Master key, followed by imprinting the new external split on a purple PED Key - any pre-existing SRK becomes invalid.

Have available:

- your SafeNet appliance
- the SafeNet PED and cable
- two "new" purple keys [either they are new and blank from the factory, or they have been used for some other purpose (could include previous use as purple keys for another HSM) that do not contain a currently valid SRV for this HSM - we suggest that you apply purple labels before you start]

Perform the SRK enable operation, and enter Secure Transport Mode now:

1. Connect the SafeNet PED to the SafeNet Network HSM appliance.
2. Login to the appliance as 'admin'.
3. Login to the HSM

```
hsm login
```

4. Run the command

```
hsm srk enable
```

Follow the PED prompts, introducing the purple key (*) and pressing buttons on the PED keypad.

Record the SRV verification string when it is presented.

This command performs a resplit of the MTK before moving one of the (new) splits out to your purple PED Key(s).

5. Run the command

```
hsm srk transportMode enter
```

Follow the PED prompts, presenting the purple key from step 4 (above) when prompted.

6. When the HSM enters Secure Transport Mode, shut it down with
`sysconf appliance poweroff`
7. Disconnect the appliance and pack it for shipment.
8. Ship the imprinted purple SRK to your customer or remote location, separately from the appliance shipment.
9. Send the STM verification string via another path.

(* For simplicity, this procedure assumes that you choose Mvalue and Nvalue as "1" in order to have a single imprinted purple key. You can choose other values for M and N, to split the SRV across multiple purple PED Keys. Similarly, you are also given the opportunity to make a copy of the purple SRK - or of the MofN group of purple PED Keys if you elected to use MofN. Your choices in these matters are dictated by convenience and by your security policies.)

RECOVER

At the new location, unpack the appliance and connect it to power, network, and PED.

You should also have available the PED Keys for that HSM, including the purple PED Key (*) that you received by separate shipment, as well as the verification string.

10. Power up the appliance and log in.
11. Compare the verification string that was shipped with the string that is presented when you run the command
`hsm srk keys verify`
 The PED prompts for the SRK (purple PED Key) and shows the verification string. The displayed string should match the string that was sent by mail or courier.
12. Run the command
`hsm srk transportMode recover`
 Respond to the PED prompts (*).

You can now use the HSM's blue PED Key to log in and administer the HSM, and black PED Keys to activate partitions for clients to access.

(*If you invoked MofN when creating the SRK, you must provide quantity M of the purple keys containing the SRV splits.)

No Re-split?

This section describes the same procedure if SRK was previously enabled, and you wish to *re-use* the existing purple PED Key. This might be the case if you know that the most recent re-split of the master key was performed by your organization.

Have available:

- your SafeNet appliance
- the SafeNet PED and cable
- the existing, imprinted purple PED Key[s]
- the verification string for the existing purple key [because you are not re-splitting or enabling at this time, the verification string for the Secure Recovery Vector on the existing key is not presented, so you must already have it in your possession]

Perform the SRK enable operation, and enter Secure Transport Mode now:

1. Connect the SafeNet PED to the SafeNet Network HSM appliance.
2. Login to the appliance as 'admin'.

3. Run the command

```
hsm srk transportMode enter
```

Follow the PED prompts, presenting the current purple PED Key (or presenting quantity M of the purple PED Keys, if you had invoked MofN during the most recent enable or resplit) when prompted.

4. When the HSM enters Secure Transport Mode, shut it down with

```
sysconf appliance poweroff
```

5. Disconnect the appliance and pack it for shipment.

6. Ship the imprinted purple SRK to your customer or remote location, separately from the appliance shipment.

7. Send the STM verification string via another path.

(* For simplicity, this procedure assumes that you choose Mvalue and Nvalue as "1" in order to have a single imprinted purple key. You can choose other values for M and N, to split the SRV across multiple purple PED Keys. Similarly, you are also given the opportunity to make a copy of the purple SRK - or of the MofN group of purple PED Keys if you elected to use MofN. Your choices in these matters are dictated by convenience and by your security policies.)

Additional Notes

Re-split?

You do not need to re-split if you have recently created a new purple PED Key for this HSM (as happens when you `hsm srk enable`), or if you have maintained rigorous records and are confident that your purple PED Key has not been compromised.

To be absolutely sure, we recommend that you perform a re-split before placing the HSM in Secure Transport Mode, unless your procedures require that you use an existing purple PED Key.

Number of purple keys

The basic procedures above suggest that you need either one or two purple PED Keys.

However, choices that you can make while interacting with the SafeNet PED could require several additional purple keys.

Copies?

During every imprinting transaction, the PED asks if you wish to copy the inserted PED Key. Therefore, ensure that you have enough spare/blank keys ready for the number of copies that you expect to make (examples: alternate key holder, on-site stored back-up, off-site back-up/escrow).

MofN?

At the beginning of an imprinting operation, the PED asks for values of "M" and "N". This is for "MofN" split-secret multi-user (or multi-part) authentication, explained elsewhere in this manual (see ["Using MofN" on page 343](#)). If you input "1" for both "M" and "N", then MofN is not in operation, and only the single (in this case purple) PED Key is needed to carry the authentication secret. But if you specify numbers higher than "1", then the PED splits the secret and imprints the splits onto quantity N different keys. Later, any operation that calls for that secret will need quantity M of those splits to be re-united (presented to the PED upon demand). Thus you must have quantity "N" spare/blank keys ready if you intend to invoke MofN.

Copies and MofN?

Furthermore, you might wish to combine the two scenarios above - in most situations it is prudent to have a backup of any authentication device, against loss or damage. If that is your policy, and if you intend to invoke MofN, then you will need enough spare/blank keys to make at least two full MofN sets of the SRK (purple PED Key(s)). In that case, be careful not to mix the copy sets. If you chose MofN as 3 of 5, then the 5 keys get 5 different splits of the purple-key

secret. When you later need to present the secret, you need 3 different splits. If the sets are accidentally mixed, you could have three key holders standing at the PED, but with two of their keys being duplicate splits - the PED refuses to accept what it sees as the same key twice when recreating an MofN secret. To avoid the possibility of inadvertently mixing copies and originals, one option is to simply choose a small "M" and a large "N", and not make copies. For example, if you chose M=3 and N=15, you could arbitrarily select three groups of five splits and use them as though the split was "3 of 5". That is, you could distribute 5 of the 15 splits to operations personnel, so that any three of them could authenticate to the HSM. You could keep another 5 of the 15 splits in on-site lockup as on-site backup, and you could keep the third group of 5 of the 15 splits as your off-site backup. In that case, you avoid the possibility of accidentally mixing copied smaller groups, which could result in two or three of the same split in one group - which the PED would reject.

View a table that compares and contrasts various "deny access" events or actions that are sometimes confused ("[Comparison of Destruction/Denial Actions](#)" on page 400).

Secure Transport Mode [Remote]

This topic describes what to do if you wish to invoke Secure Transport Mode (STM) on a remote SafeNet Network HSM, when shipping the appliance:

- to your customer or
- to your partner organization or
- to your own personnel at another site within your organization,

That is, as the appliance administrator and the HSM Admin or SO, you are not present when Secure Transport Mode is invoked and the appliance is packed for shipment, and you are not present at its destination when the appliance is unpacked and readied for use.

On-site technical personnel are performing the physical take-down, packing, unpacking and setup, but you remain at your remote location, administering the appliance and HSM via SSH and controlling access via Remote PED.

You could also use STM for securely storing the HSM, where "transport" would take place simply into, and later out of, your warehouse or vault. However, you would also need to manage separate secure storage and handling of the imprinted purple PED Key (SRK) for that HSM until it was time to recover the HSM and return it to service.

This page applies to PED Authenticated HSMs only. It does not apply to Password Authenticated HSMs.

This page assumes that you have a remote-capable SafeNet PED 2 (Remote Capable), and associated pedserver.exe software installed on your local-to-you computer.

You have already set up the SafeNet Network HSM for Remote PED operation, before you shipped it to its current remote location - that is, you imprinted the HSM and an orange PED Key with the Remote PED Vector (RPV), and you have that orange key available.



CAUTION: If the HSM contents are of any value, perform backups of your partitions before you continue with Secure Transport Mode procedure. See "[Remote Application-Partition Backup and Restore Using the Backup HSM](#)" on page 68 for more information.

Make a Remote PED Connection

First, using an ssh session, display the current status of the remotely located SafeNet Network HSM, to know your starting point.

```
[192.168.9.72] lunash:>hsm ped show
Ped Client Version 1.0.5 (10005)
```

```
Ped Client launched in status mode.
Ped PedClient is not currently running.
Show command passed.
Command Result : 0 (Success)
[192.168.9.72] lunash:>
```

Start pedServer.exe on your local computer.

Via SSH, tell the Remote PED Client on the SafeNet Network HSM to find and connect to the PED Server (pedServer.exe) on the selected computer - most likely the computer where you are currently working.

```
[192.168.9.72] lunash:>hsm ped connect -ip 192.168.10.175 -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.
Starting background process
Background process started
Ped Client Process created, exiting this process.
Command Result : 0 (Success)
[192.168.9.72] lunash:>
```

Confirm that the link is established.

```
[192.168.9.72] lunash:>
[192.168.9.72] lunash:>hsm ped show
Ped Client Version 1.0.5 (10005)
Ped Client launched in status mode.
Ped Client is connected to a Ped Server.
```

Client Information

```
Hostname: 192.168.9.72
IP: 192.168.9.72/192.168.254.254
Firmware Version: 6.0.7
HSM Cmd Protocol Version: 15
Callback IO Version: 1
Callback Protocol Version: 1
Software Version: 1.0.5 (10005)
```

Server Information

```
Hostname: OTT1-202011
IP: 192.168.10.175
Firmware Version: 2.4.0-3
PedII Protocol Version: 1.0.1-0
Software Version: 1.0.5 (10005)
Ped2 Connection Status: Connected
Ped2 RPK Count: 1
Ped2 RPK Serial Numbers: (70540100834a2301)
```

Operating Information

```
Server Port: 1503
Admin Port: 1501
External Admin Interface: No
Client Up Time: 31 (secs)
Client Current Idle Time: 7 (secs)
Client Total Idle Time: 9 (secs) (29%)
Idle Timeout Value: 1800 (secs)
```

```
Show command passed.
Command Result : 0 (Success)
[192.168.9.72] lunash:>
```

Check SRK status

```
[192.168.9.72] lunash:>hsm srk show
Secure Recovery State flags:
=====
External split enabled:      no
SRK resplit required:       no
Hardware tampered:          no
Transport mode:              no
Command Result : 0 (Success)
```

Enable SRK

```
192.168.9.72] lunash:>hsm srk enable
Luna PED operation required to enable external SRK split - use Secure Recovery (purple) PED key.
```

In RemotePED, answer the following prompts:

```
M value (1-16)
N value (M-16)
Insert a SRK PED key and press ENTER
This PED Key is for SRK, overwrite? Yes/No
**warning** Are you sure you want to overwrite this PED Key? Yes/No
Enter new PED PIN:
Confirm new PED PIN:
Are you duplicating this keyset? (Y/N)
```

PED shows “STM Enabled”

```
Command Result : 0 (Success)
[192.168.9.72] lunash:>hsm srk show
Secure Recovery State flags:
=====
External split enabled:      yes
SRK resplit required:       no
Hardware tampered:          no
Transport mode:              no
Command Result : 0 (Success)
```

Enter Secure Transport Mode

```
[192.168.9.72] lunash:>hsm srk transportMode enter
CAUTION: You are about configure the HSM in transport mode.
If you proceed, the HSM will be inoperable until it
is recovered with the Secure Recovery Key.
Type 'proceed' to continue, or 'quit' to quit now.
> proceed
Configuring the HSM for transport mode...
Luna PED operation required to enter transport mode - use Secure Recovery (purple) PED key.
Be sure to record the verification string that is displayed after the MTK is zeroized.
```

In RemotePED, answer following prompts:

```
Insert a SRK PED key and press ENTER
Generating a verify string ECKS-W7xT-Ep9E-psGb, Continue? (Y/N)
```

PED shows “SRK was zeroized”

HSM is now in Transport Mode.


```

Command Result : 0 (Success)
[192.168.9.72] lunash:>hsm srk show
Secure Recovery State flags:
=====
External split enabled:      yes
SRK resplit required:       no
Hardware tampered:          no
Transport mode:              yes
Command Result : 0 (Success)

```

At this point, pack the HSM appliance and ship to your eventual recipient via the most secure means (courier) available.

The options now are:

- You keep the purple PED Key and the verification string and ship only the HSM - you will perform the recovery from your administrative location, once the HSM is installed at the remote location. This would be the situation if you were shipping within your organization and retaining control centrally, or if you were shipping to a customer who is leasing the equipment, but you are retaining ultimate administrative control.
OR
- You have remotely configured and administered the HSM, while personnel at your own remote location did the physical work to make connections, then they disconnected the HSM when you finished accessing it, and packed it for shipment. From that transshipment point, the HSM is now being forwarded to your customer, who will take over complete responsibility.

If you keep control

In the first scenario, you retain all PED Keys and will perform further administrative actions from your location when the HSM reaches its new destination - you retain control; you manage the physical security of the purple PED Key and the verification string, which you will use when you perform STM recovery remotely (below).

The subsequent instructions on this page assume this scenario, where you have remotely set the HSM into Secure Transport Mode, and you will be remotely taking the HSM out of Secure Transport Mode, once it has arrived at its next location and been set up.

If you transfer control

In the second scenario, you relinquish administrative control of the HSM, so you ship the purple PED Key and the verification string to the eventual owners/administrators of the HSM.

- Send the HSM to your recipient by the most secure means available.
- Send the purple PED Key, from the above steps, to your recipient via a different carrier (courier, post, other).
- Send the verification string that you just recorded (above) to your recipient by yet another means.

In this way, you are ensuring that the three components (HSM, purple PED Key, and verification string for that specific PED Key) cannot be brought together between the time they leave your hands and the time that they arrive (separately) at the recipient destination.

In this scenario, your recipient should also have this Help, and they can decide whether to use the local instructions or the remote instructions (below) to bring the received HSM out of Secure Transport Mode.

What if someone makes a new SRK while the HSM is in Transport Mode?

The HSM refuses to allow such action. Here is an example of an attempt, and the result.

SRK Resplit (attempt) while HSM is in Transport Mode

```
[192.168.9.72] lunash:>hsm srk keys resplit
```

Error: The Secure Recovery Key cannot be resplit when the HSM is in transport mode or tampered. Use the recover command to restore the HSM to a functional state.
 Error: 'hsm srk keys resplit' failed. (C0000400 : RC_TOKEN_STATE_INVALID)
 Command Result : 65535 (Luna Shell execution)

SRK Key verify (attempt) while HSM in Transport Mode

```
[192.168.9.72] lunash:>hsm srk keys verify
Error: The SRK cannot be verified when the HSM is in transport mode
or tampered. Use the recover command to restore the
HSM to a functional state.
Error: 'hsm srk keys verify' failed. (C0000400 : RC_TOKEN_STATE_INVALID)
Command Result : 65535 (Luna Shell execution)
```

At the destination, recover from Secure Transport Mode

```
[192.168.9.72] lunash:>hsm srk transportMode recover
Attempting to recover from Transport Mode...
Luna PED operation required to recover the HSM - use Secure Recovery (purple) PED key.
In RemotePED, respond to the following prompts as appropriate:
```

```
Insert a
SRK PED key and
press ENTER
Generating a verify string
ECSK-W7xT-Ep9E-psGb,
Continue? (Y/N)
```

Luna PED shows “SRK was restored” and lunash command line shows:

```
Successfully recovered from transport mode.
HSM restored to normal operation.
Command Result : 0 (Success)
[192.168.9.72] lunash:>hsm srk show
Secure Recovery State flags:
=====
External split enabled:      yes
SRK resplit required:       no
Hardware tampered:          no
Transport mode:             no
Command Result : 0 (Success)
```

SRK key resplit

Having received and unlocked your HSM, you might now prefer to invalidate the current SRK and create a new external split for future use.

```
[192.168.9.72] lunash:>hsm srk keys resplit
Luna PED operation required to resplit the SRK - use Secure Recovery (purple) PED key.
In RemotePED, answer following question accordingly:
Insert a SRK PED key and press ENTER
M value (1-16)
N value (M-16)
Insert a
SRK PED key and
press ENTER (insert old SRK key here)
```

This PED Key is for SRK,
 overwrite? Yes/No

Note, you see the above message if the key that you present has previously been imprinted with a Secure Recovery Vector.

```
**warning** Are you sure you want to overwrite this PED Key? Yes/No
Enter new PED PIN:
Confirm new PED PIN:
Are you duplicating this keyset? (Y/N)
Ped shows "SRK was resplit"
SRK resplit succeeded.
Command Result : 0 (Success)
[192.168.9.72] lunash:>hsm srk show
Secure Recovery State flags:
=====
External split enabled:      yes
SRK resplit required:      no
Hardware tampered:         no
Transport mode:            no
Command Result : 0 (Success)
```

Verify the new SRK

```
[192.168.9.72] lunash:>hsm srk keys verify
Luna PED operation required to verify the SRK split - use Secure Recovery (purple) PED key.
On the Remote PED, respond to the prompts:
```

```
Insert a SRK PED key and press ENTER
PED shows "SRK was restored"
SRK verified.
Command Result : 0 (Success)
[192.168.9.72] lunash:>hsm srk show
Secure Recovery State flags:
=====
External split enabled:      yes
SRK resplit required:      no
Hardware tampered:         no
Transport mode:            no
Command Result : 0 (Success)
```

SRK disable

This section shows how to disable SRK - returning the external split (Secure Recovery Vector) of the Master Key from its location on the external purple PED Key to a location inside the HSM. After this action, Secure Transport Mode is not possible unless you Enable again. Also, with the two recovery splits held inside the HSM, the HSM can recover from a physical tamper event with only a reboot.

```
[192.168.9.72] lunash:>hsm srk disable
Luna PED operation required to disable external SRK split - use Secure Recovery (purple) PED key.
```

In RemotePED, respond to the following prompts:

```
Insert a
SRK PED key and
press ENTR
```

SafeNet PED shows "STM Disabled"

```
Command Result : 0 (Success)
[192.168.9.72] lunash:>hsm srk show
```

```
Secure Recovery State flags:
=====
External split enabled:      no
SRK resplit required:       no
Hardware tampered:          no
Transport mode:              no
Command Result : 0 (Success)
[192.168.9.72] lunash:>
```

Re-Split Required

The "SRK resplit required" flag is set only in the event that a failure occurred during a re-split operation, leaving the HSM in an intermediate state. An example might be the user pressing cancel at the wrong time, or a power failure or disconnection during a re-split.

```
Secure Recovery State flags:
=====
External split enabled:  yes
SRK resplit required:  yes
Hardware tampered:      no
Transport mode:         no
```

After an incomplete hsm srk resplit, an attempt to login or to perform other HSM operations would yield an error message about the MTK state. The HSM would process only view/show commands while in that state.

In that situation, it is operationally urgent to issue the command:

```
hsm srk keys resplit
```

which creates a new split of the SRK and places the external portion on a new purple PED Key (or keys, if you choose to invoke MofN).

The HSM is once more usable.

Security

An attacker lacking the proper purple key cannot place the HSM into "SRK resplit required" state. Only the holder of the legitimate purple PED Key can start an hsm srk resplit operation, and is therefore entitled to resume/restart that operation if it is interrupted.

Slot Numbering and Behavior

Administrative partitions and application partitions are identified as PKCS#11 cryptographic slots in SafeNet utilities, such as LunaCM and multitoken, and for applications that use the SafeNet library.

Order of Occurrence for Different SafeNet HSMs

A host computer with SafeNet HSM Client software and SafeNet libraries installed can have SafeNet HSMs connected in any of three ways:

- PCI-e embedded/inserted SafeNet PCI-E HSM card (one or multiple HSMs installed - administrative partitions and application partitions are shown separately if HSM firmware is version 6.22.0 or newer)
- USB-connected SafeNet USB HSMs (one or multiple - administrative partitions and application partitions are shown separately if HSM firmware is version 6.22.0 or newer)
- SafeNet Network HSM application partitions(*), registered and connected via NTLS or via STC.

Any connected HSM partitions are shown as numbered slots. Slots are numbered from zero or from one, depending on configuration settings (see "[Settings Affecting Slot Order](#)" on the next page, below), and on the firmware version of the HSM(s).

(*One or multiple application partitions. Administrative partitions on SafeNet Enterprise HSMs are not visible via lunacm and other client-side tools. Only registered, connected application partitions are visible, of which multiple-per-HSM, up to 100, can exist. That is, a remote SafeNet Network HSM might support 100 application partitions, but your application and lunacm might see only one or two or fifteen of them if those were the only ones that had established certificate-exchange NTLS links with the current Client computer.)

In lunacm, a slot list would normally show:

- SafeNet Network HSM application partitions for which NTLS links are established with the current host, followed by
- SafeNet PCI-E HSM cards, followed by
- SafeNet USB HSMs

For SafeNet Network HSM, as seen from a client (via NTLS), only application partitions are visible. The HSM administrative partition of a remote SafeNet Network HSM is never seen by a SafeNet HSM Client. The SafeNet Network HSM slots are listed in the order they are polled, dictated by the entries in the [SafeNet Network HSM] section of the Crystoki.ini / chrystoki.conf file, like this:

```
ServerName00=192.20.17.200
ServerPort00=1792
ServerHtl00=0
ServerName01=192.20.17.220
ServerPort01=1793
ServerHtl01=
```

For SafeNet PCI-E HSM and SafeNet USB HSM, if you have multiple of either HSM type connected on a single host, then the order in which they appear is the hardware slot number, as discovered by the host computer.

For SafeNet PCI-E HSM and SafeNet USB HSM, the HSM administrative slot always appears immediately after the application partition. If no application partition has yet been created, a space is reserved for it, in the slot numbering.

Settings Affecting Slot Order

Settings in the [Presentation] section of the configuration file (Chrystoki.conf for UNIX/Linux, crystoki.ini for Windows) can affect the numbering that the API presents to SafeNet tools (like lunacm) or to your application.

[Presentation]

ShowUserSlots=<slot>(<serialnumber>)

- Sets starting slot for the identified partition.
- Default, when ShowUserSlots is not specified, is that all available partitions are visible and appear in default order.
- Can be applied, individually, to multiple partitions, by a single entry containing a comma-separated list like:
ShowUserSlots=1(351970018022),2(351970018021),3(351970018020),....
- Affects only PPSO partitions (f/w 6.22.0 or newer)
- If multiple partitions on the same HSM are connected to the SafeNet HSM Client host computer, redirecting one of those partitions with ShowUserSlots= causes all the others to disappear from the slot list, unless they are also explicitly re-ordered by the same configuration setting.

ShowAdminTokens=yes

- Default is yes. Admin partitions of local HSMs are visible in a slot listing.
- Remotely connected partitions (SafeNet Network HSM) are not affected by this setting, because NTLS connects only application partitions, not HSM SO (Admin) partitions to clients, so a SafeNet Network HSM SO administrative partition would never be visible in a client-side slot list, regardless.

ShowEmptySlots=1

- Controls how C_GetSlotList - as used by lunacm slot list command, or ckdemo command 14, and by your PKCS#11 application - displays, or does not display unused potential slots, when the number of partitions on an HSM is not at the limit.

OneBaseSlotId=1

- Causes basic slot list to start at slot number 1 (one) instead of default 0 (zero).
(Any submitted number other than zero is treated as "1". Any letter or other non-numeric character is treated as "0".)

Effects of Settings on Slot List

Say, for example, you have multiple HSMs connected to your host computer (or installed inside), with any combination of firmware 6.22.0 (and newer) or pre-6.22.0 firmware, and no explicit entries exist for slot order in the config file. The defaults prevail and the slot list would start at zero.

If you set OneBaseSlotId=1 in the configuration file, then the slot list starts at "1" instead of at "0". You could set this for personal preference, or according to how your application might expect slot numbering to occur (or if you have existing scripted solutions that depend on slot numbering starting at zero or starting at one). OneBaseSlotId affects the starting number for all slots, regardless of firmware.

If you set `ShowUserSlots=20(17923506)`, then the identified token or HSM or application partition would appear at slot 20, regardless of the locations of other HSMs and partitions, but only if the indicated partition is firmware 6.22.0 or newer and is a PPSO partition.

Effects of New Firmware on Slot Login State

Note: Slots retain login state when current-slot focus changes.



For HSMs with firmware earlier than version 6.22.0, when you used **slot set** to move the focus from an HSM partition or slot with logged in session(s), to another partition or slot, any sessions on the original slot were automatically closed (thus logged out).

For HSMs with firmware version 6.22.0 or newer, you can use **slot set** to repeatedly shift focus among slots, and whatever login state was in force when you were previously focused on a slot is still in effect when you return to that slot.

SNMP Monitoring

This chapter describes Simple Network Management Protocol (SNMP v3) support for remote monitoring of conditions on a local HSM that might require administrative attention. It contains the following sections:

- "Overview and Installation" below
- "The SafeNet Chrysalis-UTSP MIB" on page 438
- "The SAFENET HSM MIB" on page 439
- "Frequently Asked Questions" on page 446

Overview and Installation

This section provides an overview of the SNMP implementation and describes how to install the SNMP subagent.

MIB

We provide the following MIBs (management information base):

MIB Name	Description
CHRYSLIS-UTSP-MIB.txt	Defines SNMP access to information about the SafeNet appliance.
SAFENET-HSM-MIB.txt	Defines SNMP access to information about the SafeNet HSM.
SAFENET-GLOBAL-MIB.txt	Must be found in your system path so that symbols can be resolved.
SAFENET-APPLIANCE-MIB.txt	Reports the software version of SafeNet Network HSM appliance.

Copy all MIBs in <luna client install dir> to the MIB directory on your system.

For SafeNet Network HSM, the host is the appliance, so all the above MIBs are in the appliance, to support SNMP.

SafeNet SNMP Subagent

We find that most customers choosing to use SNMP already have an SNMP infrastructure in place. Therefore, we provide a subagent that you can install on your managed workstations, and which can point to your agent via the socket created by the agent. This applies to SafeNet USB HSM and SafeNet PCI-E HSM - for SafeNet Network HSM, the subagent is already on the appliance.

The SNMP subagent (luna-snmp) is an AgentX SNMP module that extends an existing SNMP agent with support for SafeNet HSM monitoring. It is an optional component of the SafeNet HSM client installation. The subagent has been tested against net-snmp, but should work with any SNMP agent that supports the AgentX protocol.

To install the SNMP subagent

After selecting one or more products from the main SafeNet HSM Client installation menu, you are presented with a list of optional components, including the SNMP subagent. It is not selected by default, but can be installed with any product except the SafeNet Network HSM client installed in isolation.

1. In the installation media, go to the appropriate folder for your operating system.
2. Run the installer (install.sh for Linux and UNIX, LunaClient.msi for Windows).
3. Choose the SafeNet products that you wish to install, and include SNMP among your selections. The subagent is installed for any SafeNet product except SafeNet Network HSM in isolation.
4. Proceed to Post-installation configuration.

Post-installation configuration

After the SafeNet HSM client is installed, complete the following steps to configure the SNMP subagent:

1. Copy the SafeNet MIBs from <install dir>/snmp to the main SNMP agent's MIB directory. Or copy to another computer (your SNMP computer) if you are not running SNMP from the same computer where SafeNet Client software is installed.
2. If running on Windows, configure the subagent via the file <install dir>/snmp/luna-snmpp.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-snmpp.conf.

3. After configuration is complete, start the agent. Then start the subagent via the service tool applicable to your platform (for example, "service luna-snmpp start" on Linux, or start SafeNet SNMP Subagent Service from the services in Windows).

Normally the agent is started first. However, the subagent periodically attempts to connect to the agent until it is successful. The defaults controlling this behavior are listed below. They can be overridden by changing the appropriate entries in **luna-snmpp.conf**.

Configuration Options In the luna-snmpp.conf File

Option	Description	Default
agentXSocket [<transport-specifier>:] <transport-address> [,...]	Defines the address to which the subagent should connect. The default on UNIX-based systems is the Unix Domain socket "/var/agentx/master". Another common alternative is tcp:localhost:705. See the section LISTENING ADDRESSES in the snmpd man page for more information about the format of addresses (http://www.net-snmp.org/docs/man/snmpd.html).	The default, for Linux, is "/var/agentx/master". In the file, you can choose to un-comment "tcp:localhost:705" which is most commonly used with Windows.
agentXPingInterval <NUM>	Makes the subagent try to reconnect every <NUM> seconds to the master if it ever becomes (or starts) disconnected.	15

Option	Description	Default
agentXTimeout <NUM>	Defines the timeout period (NUM seconds) for an AgentX request.	1
agentXRetries <NUM>	Defines the number of retries for an AgentX request.	5

The SafeNet Chrysalis-UTSP MIB



Note: The Chrysalis MIB is the SafeNet MIB for all SafeNet HSM products - the Chrysalis name is retained for historical continuity.

To illustrate accessing data, the command "snmpwalk -v 3 -u admin -l authPriv -a SHA1 -A 12345678 -x AES -X 87654321 myLuna19 private" produced this output:

- CHRYSALIS-UTSP-MIB::hsmOperationRequests.0 = Counter64: 3858380
- CHRYSALIS-UTSP-MIB::hsmOperationErrors.0 = Counter64: 385838
- CHRYSALIS-UTSP-MIB::hsmCriticalEvents.0 = Counter64: 0
- CHRYSALIS-UTSP-MIB::hsmNonCriticalEvents.0 = Counter64: 5
- CHRYSALIS-UTSP-MIB::ntlsOperStatus.0 = INTEGER: up(1)
- CHRYSALIS-UTSP-MIB::ntlsConnectedClients.0 = Gauge32: 0
- CHRYSALIS-UTSP-MIB::ntlsLinks.0 = Gauge32: 0
- CHRYSALIS-UTSP-MIB::ntlsSuccessfulClientConnections.0 = Counter64: 16571615927115620
- CHRYSALIS-UTSP-MIB::ntlsFailedClientConnections.0 = Counter64: 1657161592711562

The various counts are recorded since the last restart.

Item	Description
hsmOperationRequests	The total number of HSM operations that have been requested.
hsmOperationErrors	The total number of HSM operations that have been requested, that have resulted in errors.
hsmCriticalEvents	The total number of critical HSM events that have been detected (Tamper, Decommission, Zeroization, SO creation, or Audit role creation)
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
```

The information is defined in tables, as detailed in the following sections:

SNMP Table Updates

The SNMP tables are updated and cached every 60 seconds. Any changes made on the HSM may therefore take up to 60 seconds to be included in the tables. When a query is received to view the tables, the most recent cached version is displayed. If a change you were expecting is not displayed, wait 60 seconds and try again.



Note: Some values may not get updated automatically, such as the HSM firmware version (hsmFirmwareVersion) following a firmware upgrade. To force an update, restart the SNMP agent.

hsmTable

This table provides a list of all the HSM information on the managed element.

Item	Type	Description	Values
hsmSerialNumber	DisplayString	Serial number of the HSM - used as an index into the tables.	From factory
hsmFirmwareVersion	DisplayString	Version of firmware executing on the HSM.	As found
hsmLabel	DisplayString	Label associated with the HSM.	Provided by SO at init time
hsmModel	DisplayString	Model identifier for the HSM.	From factory
hsmAuthenticationMethod	INTEGER	Authentication mode of the HSM.	unknown(1), -- not known password(2), -- requires

Item	Type	Description	Values
			passwords pedKeys(3) -- requires PED
hsmRpvInitialized	INTEGER	Remote ped vector initialized flag of the HSM.	notSupported(1), -- rpv not supported uninitialized(2), -- rpv not initialized initialized(3) -- rpv initialized
hsmFipsMode	TruthValue	FIPS 140-2 operation mode enabled flag of the HSM.	Factory set
hsmPerformance	INTEGER	Performance level of the HSM.	
hsmStorageTotalBytes	Unsigned32	Total storage capacity in bytes of the HSM	Factory set
hsmStorageAllocatedBytes	Unsigned32	Number of allocated bytes on the HSM	Calculated
hsmStorageAvailableBytes	Unsigned32	Number of available bytes on the HSM	Calculated
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	Counted

Item	Type	Description	Values
		reset	

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 available (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)

Item	Type	Description	Values
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
hsmClientOTTEpiry	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 SafeNet tools output

For comparison, the following shows lunacm or lunash command outputs that provide HSM information equivalent to the SNMP information depicted in the tables above (from the HSM MIB).

HSM Information

At the HSM level the information in the outputs of "hsm show" and "hsm 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 be 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
Enable private key SFF backup/restore	Disallowed
Enable secret key SFF backup/restore	Disallowed
Enable Secure Trusted Channel	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
Force Secure Trusted Channel	Off	37

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?

Software Maintenance and Updates

This chapter describes how to maintain and update the functionality of your HSMs by performing the following tasks:

- "About Updating SafeNet Network HSM" below
- "Advanced Configuration Upgrades" on page 451
- "Applying SafeNet HSM Capability Upgrades" on page 455

About Updating SafeNet Network HSM

Your SafeNet Network HSM system consists of components that might, from time to time, require updating to newer versions. The newer version might have fixes or functional improvements that are useful or important for your application. The components that might be affected are:

- Client software
- SafeNet Network HSM system software (the lunash command-line shell and its underlying software)
- SafeNet Network HSM keycard firmware
- SafeNet Backup HSM firmware

How Firmware Updates Affect Agency Validation

In the case of FIPS 140, cryptographic devices are evaluated as a combination of hardware and firmware. Therefore, if either of those elements changes, the device is no longer covered by the current validation certificate. If you require that equipment used in your application be (for example) FIPS 140-2 level 3 validated, you can use the most recent of our relevant HSM products that has been validated - which applies to a specific hardware and firmware combination. If we release a newer version of firmware, your own security or compliance policies would not permit you to install that update until we have submitted the updated HSM for [re-] evaluation, and a new validation certificate has been issued.

As a general rule (exceptions are possible) we submit HSMs with new firmware versions. If the changes are small or do not affect areas that concern the FIPS evaluators, then the re-evaluation is performed on a delta basis and therefore occurs relatively quickly. For a completely new product or major revision, the evaluators require a complete re-submission and the process takes roughly a year from submission to certificate. Therefore, when a FIPS-candidate firmware version exists, our practice is to ship the respective HSM product with the most recent FIPS-validated firmware version installed, and with the candidate version as a standby update file (on the appliance, ready to install, but not yet installed). This ensures that customers who require validated systems continue to get them, and that customers who do not require validated systems are able to easily and quickly apply the update if they choose to do so.

The obvious trade-off is that customers who elect to remain with the as-shipped installed firmware version are maintaining the FIPS compliance at the cost of any upgraded capabilities or any security or functional fixes that are part of the firmware update. Similarly, customers who choose to perform the update benefit from the improved capabilities and any security or functional fixes, but at the cost of moving out of FIPS compliance.

Updating the Client Software

To update the software on a Client, you simply remove the older version and Install the newer, using the same procedure (for your operating system) that you used for the original software installation. That applies to SafeNet Network HSM Client software itself, as well as to the SDK material.

As an example, the Client uninstall, when invoked on Windows, removes libraries, utilities and other material related to the client, but does not remove configuration files and certificates. This allows you to install the newer version and be able to resume operation without need to manually restore configuration settings and without need to recreate, re-exchange, and re-register client and appliance certificates for NTL.

Updating the Appliance Software



Note: Appliance software upgrade is a one-way operation. There is currently no way to downgrade the appliance software once a new version is applied. This contrasts with the SafeNet HSM client software, which can be replaced with any version by uninstalling the current version and installing a desired version, and the SafeNet HSM firmware, which can be rolled back to the version that was installed before the currently-installed version (applies only to versions since firmware rollback was enabled).

To update the system software

To update system software and firmware, you must move the updates, in the form of update package files, to SafeNet Network HSM and apply them. Updates are accompanied by instructions that provide detailed update instructions for each component. System and firmware updates require an authentication code, which is provided in a text file accompanying the update package.

1. Copy the SafeNet Network HSM 6.x.y-z Appliance package file from the ftp directory to the SafeNet Network HSM, as follows:
`scp \<path>\lunasa_update-6.x.y-z.spkg admin@<LunaSAhostname>:`
 where x.y-z is the version and build number
 (in Windows, use the supplied PSCP utility).
2. Stop all client applications connected to the SafeNet Network HSM appliance.
3. At the shell prompt, log in to the SafeNet Network HSM appliance as admin.
4. Log in to the SafeNet Network HSM as HSM Admin or SO.
 Use `lunash:>hsm login`
 For SafeNet Network HSM with PED Authentication, the blue PED Key is required.
 For SafeNet Network HSM with Password Authentication, you are prompted for the HSM Admin (SO) password.
5. [Optional Step] Verify that the file that you copied is present on the SafeNet Network HSM
`lunash:>package listfile`
6. [Optional Step] Verify the package on the SafeNet Network HSM
`lunash:>package verify lunasa_update-6.x.y-z.spkg -authcode #####`
 where "#####" is the authorization code from the file `lunasa_update-6x.y-z.auth`.
 The verification process requires approximately one and a half minutes.
7. Install the software upgrade package on SafeNet Network HSM
`lunash:>package update lunasa_update-6.x.y-z.spkg.spkg -authcode #####`
 where "#####" is the authorization code from the file `lunasa_update-6.x.y-z.auth`.
 The installation/update process requires approximately one and a half minutes. During that time, a series of

messages shows the progress of the update.

8. At the end of this process, a message "Software update completed!" appears. The software version is now 6.x.y-z. If the software update also included a firmware update, then the firmware 6.v.w package is now on the appliance, waiting to be installed in the HSM.

Perform a reboot of the SafeNet Network HSM appliance before you update the firmware.

```
lunash:> sysconf appliance reboot
```

To update your HSM firmware to 6.v.w.

In general, a new SafeNet Network HSM is delivered with the current FIPS- validated firmware installed on the internal HSM card, and with the most recent firmware version (typically in the process of being FIPS validated) included - waiting, but not yet installed - on the SafeNet Network HSM hard drive as an optional update. Similarly, when you install a software update package that includes a firmware component, the software is changed and the accompanying new firmware goes into the waiting area on the appliance hard disk, replacing any previous optional firmware.

Regardless of whether the optional firmware update is one that was originally loaded (as an option) or one that was supplied later with a software update (as an option), it is always a separate step if you wish to install that waiting (optional) firmware into the HSM.

It is strongly recommended that your Luna SA be connected to an uninterruptible power supply (UPS) when you perform a firmware update. There is a small chance that a power failure during the update command could leave your Luna SA in an unrecoverable condition.

For PED-authenticated Luna HSMs, ensure that the SRK (the use of the purple PED Key) is disabled (bring the external portion of the MTK back into the HSM) before you begin the firmware update operation. This requires that you present the currently valid purple PED Key when you issue the **hsm srk disable** command. If you run **hsm update firmware** while SRK is enabled (a portion of the MTK is outside the HSM, on a purple PED Key) you can expect an error like:

```
Error: 'hsm update firmware' failed. (10A0B : LUNA_RET_OPERATION_RESTRICTED)
```

If you have had SRK enabled and a valid purple PED Key, you can always perform **hsm srk enable** again after the firmware update operation, and resume with a new external secure recovery vector (SRV) imprinted onto a new purple PED Key (SRK).



Note: On a multi-partition HSM, when updating from older firmware to version 6.22.0 firmware or later, you might need to re-size partitions. This is due to infrastructure that supports the Per-Partition SO (PPSO) capability, which imposes increased overhead for each partition. For additional information, see "Sizes of Partitions " on page 1.

1. Log in to the HSM with:
lunash:> hsm login
2. Run the firmware update command:
lunash:> hsm firmware upgrade
3. Log in to the HSM with:
lunash:> hsm login
4. Verify that the change has taken place (should show version 6.v.w):
lunash:> hsm show

A capability update or a firmware update is meant to be applied just one time to an HSM. If you attempt to re-apply a capability update to an HSM that already has the capability installed, the system throws an error like "C0000002 : RC_GENERAL_ERROR ". A similar result occurs if you attempt to install a particular firmware update more than once on one HSM. This is expected behavior.

For information and instructions regarding purchased Capability Updates, see "Applying SafeNet HSM Capability Upgrades" on page 455.

Standalone Firmware Update

On occasion you might need to update HSM firmware on a standalone basis - that is, where the activity is not part of appliance software update. The process is similar, except that the package transferred to the SafeNet Network HSM appliance is just a wrapper around the firmware update, and does not contain other appliance software. This can occur when you are performing evaluations and might be testing several firmware versions, or when, for whatever reason, the firmware version that you wish to install is not currently the standby firmware version on the appliance, in which case SafeNet supplies a standalone firmware update secure package .



Note: If the Secure Recovery Key (SRK) on the HSM is enabled, it must be disabled before you can upgrade the HSM firmware. The SRK is an external split of the HSM's Master Tamper Key (MTK) that is imprinted on the purple PED key. When you disable the SRK, the SRV (Secure Recovery Vector) portion of the MTK is returned to the HSM, so that the SRV is no longer external to the HSM. It is only in this state that you can upgrade the HSM firmware. After you upgrade the firmware, you can re-enable SRK, if desired, to re-imprint a purple PED key with the SRV.



Note: If you intend to re-size partitions, or to perform a firmware update (example, from pre-6.22.0 to version 6.22.0 or newer) that alters the available space in partitions, be sure to backup the contents of your HSM first. It might be required to remove some objects from partitions that are at-or-near capacity. They can be restored after all re-sizing and new-partition creation has finished.

To perform a standalone firmware update

1. Obtain the firmware update secure package from SafeNet. Use scp/pscp to upload the package to the SafeNet Network HSM appliance.

Linux/UNIX	<code>scp /<path>/<packagename>.spkg admin@<LunaSAhostname>:</code>
Windows	<code>pscp \<path>\<packagename>.spkg admin@<LunaSAhostname>:</code>

2. Stop all client applications connected to the SafeNet Network HSM appliance.
3. At the shell prompt, log in to the SafeNet Network HSM appliance as admin.
4. Log in to the SafeNet Network HSM as HSM Admin or SO.
Use `lunash:>hsm login`
For SafeNet Network HSM with PED Authentication, the blue PED Key is required.
For SafeNet Network HSM with Password Authentication, you are prompted for the HSM Admin (SO) password.
5. [Optional Step] Verify that the file that you copied is present on the SafeNet Network HSM
`lunash:>package listfile`
6. [Optional Step] Verify the package on the SafeNet Network HSM
`lunash:>package verify <packagename>.spkg -authcode #####`
where "#####" is the authorization code from the file `<packagename>.auth`.
7. Install the firmware upgrade secure package on SafeNet Network HSM
`lunash:>package update <packagename>.spkg -authcode #####`

where "#####" is the authorization code from the file <packagename>.auth.

The package update process completes in seconds. The firmware package is now on the appliance, waiting to be installed in the HSM.



Note: It is strongly recommended that your SafeNet HSM be powered from an uninterruptible power supply (UPS) when you perform the firmware update.

8. Run the firmware update command:
lunash:> hsm firmware upgrade
9. Log in to the HSM with:
lunash:> hsm login
10. Verify that the change has taken place (should show the desired target version):
lunash:> hsm show

Advanced Configuration Upgrades

SafeNet offers advanced configuration upgrades for its HSM products, some examples of which are listed in the following tables.

SafeNet delivers advanced configuration upgrades for SafeNet Network HSM as a secure package update. Follow the steps of ["Applying SafeNet HSM Capability Upgrades" on page 455](#) to apply the update.

For SafeNet PCI-E HSM and SafeNet USB HSM, you receive a firmware update file (FUF).



Note: This is not necessarily a complete list, please check with your sales representative for the full list.



Note: Part numbers shown here are for field upgrades. The same upgrades are often available for factory installation when you purchase a new SafeNet HSM product. Those have different part numbers (ask your sales representative). Not all field upgrades have an equivalent factory-applied version, because we ship HSMs with the most recent FIPS-validated firmware version, and some newer upgrades require more recent firmware, so they cannot be installed at the factory.

Table 1: SafeNet Network HSM configuration upgrades

Configuration upgrade	Part number
Maximum memory	908-000086-001
Korean algorithms	908-000139-002
ECIES acceleration	908-000175-001
5 partitions	908-000201-001
10 partitions	908-000202-001
15 partitions	908-000203-001

Configuration upgrade	Part number
20 partitions	908-000204-001
35 partitions	908-000379-001
50 partitions	908-000235-001
75 partitions	908-000280-001
100 partitions	908-000232-001
Enable Small Form-factor Backup (SA)	908-000220-001
Enable Per-Partition Security Officer (PPSO)	908-000263-001



Note: Increasing the number of partitions is not destructive; it does not erase existing partitions and objects. However, simply increasing the number of partition licenses does not increase memory. Depending on the size of the original partitions (did you re-size them to use large amounts of memory, or "all available memory"?), you might need to resize the existing partitions to make room for the additional partitions. If a partition is occupied when it is to be resized, you might need to move some objects before resizing.



Note: You can apply 100 partitions without also upgrading to Maximum Memory, but this leaves very little memory for each partition. Usefulness depends upon your application, and the sizes of keys and objects that you would store in each partition.

Also, if you are using STC, then that requires 2 KB of partition space for each STC client that is registered to a given partition.



Note: If you are both

- upgrading from an earlier firmware version to HSM firmware 6.22.0 (or newer)

AND

- applying the Per-Partition SO (PPSO) capability update,

be aware that the PPSO capability update is destructive. Therefore, there is no need to re-size partitions.

Instead, to avoid unnecessary duplication of effort, you should

- safeguard (archive) any existing partition contents,
- then zeroize the HSM for a clean update,
- then perform both the firmware AND capability updates,
- and finally restore to new partitions.

Table 2: SafeNet PCI-E HSM capability upgrades

Configuration upgrade	Part number
Korean algorithms	908-000138-002
ECIES acceleration	908-000177-001
Enable Small Form-factor Backup (PCI-E)	908-000223-001

Table 3: SafeNet USB HSM configuration upgrades

Configuration upgrade	Part number
Korean algorithms	908-000156-002
ECIES acceleration	908-000179-001

Table 4: SafeNet Backup HSM configuration upgrades

Configuration upgrade	Part number
5 partitions	908-000083-001
10 partitions	908-000287-001
20 partitions	908-000085-001
35 partitions	908-000281-001
50 partitions	908-000282-001
75 partitions	908-000283-001
100 partitions	908-000284-001

NOTE: SafeNet Remote Backup HSM comes with maximum memory and does not require a separate memory upgrade for larger numbers of partitions.

ECIES Acceleration

SafeNet offers ECIES support via a client-library shim. With the shim, ECIES 386-bit performance is approximately 40 operations per second. The ECIES acceleration configuration upgrade improves performance. This upgrade provides an approximately 5x performance increase compared to using the shim. If you choose to apply and use the configuration upgrade, you must remove the shim from your system configuration for the upgrade to have effect: shim use overrides acceleration.

Applying the ECIES advanced configuration upgrade is a destructive operation: objects already created on the HSM are destroyed. Therefore, you should apply this update when you first configure your HSM, before putting it into production (alternatively, you can back up any important objects and restore them onto the HSM after the upgrade).



Note: The full ECIES suite of mechanisms is not approved by NIST (that is, not all are FIPS 140-2 algorithms). Applying EITHER the ECIES shim OR this configuration upgrade option means that you can use all the available ECIES mechanisms when the HSM is **not** in the FIPS 140-2 mode of operation; however if FIPS 140-2 mode **is** asserted then some ECIES mechanisms are blocked.

Rollback Behavior

When it became possible to roll HSM firmware updates¹ back to an earlier version, some additional concerns became evident. The order in which you perform some activities becomes important.

An HSM that receives a firmware update arrives at that condition with any capabilities/features that were part of the HSM before the update was installed. The pre-update record of <firmware version+configuration> is set. If you rollback, you rollback² to exactly the state that was recorded, prior to the update. All the same capabilities/features would be available, because they were present before the firmware update.

Any capability that you added after a firmware update would be lost, if you then rolled back the firmware, because the pre-update record of <firmware version+configuration> did not include any capability that you added only post-update. In that case:

- If the late-installed capability **is not** dependent on the newer firmware, then you can simply install it again, on the HSM at the rolled-back firmware version, and it will become part of the pre-update record the next time you update firmware.
- If the late-installed capability **is** dependent on the newer firmware, then you must do without that feature/capability until you once more update to a firmware version that can support it. At that time, you will need to re-install that capability upgrade³.

The following table summarizes the options comparatively.

	Start with this	If you do this...	Result is this	If you next do this...	Result is this	If you next do this...	Result is this	If you next do this...	Result is this
Example 1 (Read across ==>)	f/w X and Capabilities A, B, C	Update to f/w Y	f/w Y and Capabilities A, B, C	Roll back to f/w X	f/w X and Capabilities A, B, C				

¹A newer version of client software, appliance software, or HSM firmware, to fix defects, or to improve security, or to modify/improve existing features, or to add enhancements. Updates are provided as needed, or as the product develops, for a hardware version.

²To return the HSM to its previous firmware version. This gives up any enhancements or fixes that were gained by the newer firmware version, as well as any upgrades that were installed after the firmware update (that is to be rolled back).

³A secure package that can be applied to the HSM to grant new capability or to enhance existing function.

	Start with this	If you do this...	Result is this	If you next do this...	Result is this	If you next do this...	Result is this	If you next do this...	Result is this
Example 2 (Read across ==>)		Add Capability D (no dependency)	f/w X and Capabilities A, B, C, D	Update to f/w Y	f/w Y and Capabilities A, B, C, D	Roll back to f/w X	f/w X and Capabilities A, B, C, D		
Example 3 (Read across ==>)		Update to f/w Y	f/w Y and Capabilities A, B, C	Add Capability D (no dependency)	f/w Y and Capabilities A, B, C, D	Roll back to f/w X	f/w X and Capabilities A, B, C		
Example 4 (Read across ==>)		Capability E depends on f/w Y; Attempt to add Capability E fails	f/w X and Capabilities A, B, C (unchanged)	Update to f/w Y	f/w Y and Capabilities A, B, C	Add Capability E (depends on f/w Y)	f/w Y and Capabilities A, B, C, E	Roll back to f/w X	f/w X and Capabilities A, B, C

In Example 1, above, no capabilities change; only the firmware version.

In Example 2, above, D is added **before** firmware update; therefore the pre-update record includes capability D, so **D survives** firmware update and firmware rollback.

In Example 3, above, D is added **after** firmware update, the pre-update record does not include capability D, so **D does not survive** firmware rollback.

In Example 4, above, the pre-update record does not include capability E, so E does not survive firmware rollback.

We advise you to retain a copy of any in-field configuration upgrades.

Applying SafeNet HSM Capability Upgrades

SafeNet HSMs are shipped from the factory in specific configurations with specific sets of capabilities, to suit your requirements. It can happen that your requirements change over time. To future-proof your SafeNet HSM investment, you have the option to purchase Secure Capability Updates to enhance the performance or extend the capability of SafeNet systems already in your possession, as described in "Advanced Configuration Upgrades" on page 451. The

Secure Capability Update accomplishes system upgrades while safeguarding the integrity of your sensitive key material and of the system software.

A Secure Capability Upgrade is delivered to you as a downloaded file set. The procedure to perform the update is very similar to the procedure for Appliance software updates or firmware updates.

Linux/AIX	<code>cd /usr/safenet/lunaclient/bin</code>
Solaris/HP-UX	<code>cd /opt/safenet/lunaclient/bin</code>
Windows	<code>cd C:\Program Files\SafeNet\LunaClient</code>
Linux/UNIX	<code>./scp /<path>/<spkg_patch_file.spkg> admin@<LunaHostname>:</code>
Windows	<code>pscp \<path>\<spkg_patch_file.spkg> admin@<LunaHostname>:</code>

Preparing to Upgrade

To ensure a trouble-free installation, you must prepare for the upgrade.

To prepare for the upgrade

1. Backup application partitions to SafeNet Backup HSM or Tokens (if you have the backup option).
2. On the host computer, acquire the capability update software files.
 - a. Follow the FTP instructions that are supplied in e-mail from SafeNet Customer Support (support@safenet-inc.com).
 - b. Unzip the files (as directed in the ftp instructions).

In some Windows configurations, you might not have authority to copy or unzip files directly into C:\Program Files\.... In that case, put the files in a known location that can be referenced in a lunacm command.

Installing the Upgrade Package

Once the files are unpacked and available on the host computer, open a command-prompt session.

To install the upgrade package

1. Go to the SafeNet Client directory and launch lunacm.
2. Log into the HSM:

For HSM with pre-6.22.0 firmware

```
lunacm:> hsm login
```

For HSM with version 6.22.0 or newer firmware

```
lunacm:> role login -name Administrator
```

3. Apply the new capability:

```
lunacm:>hsm updatecap -cuf \Users\me\Downloads\621-000099-001.CUF -authcode \User-
s\me\Downloads\621-000099-001_authcode.TXT
```

```
You are about to apply a destructive update.
All contents of the HSM will be destroyed.
```

All partition roles will be destroyed.
The domain will be destroyed.

Are you sure you wish to continue?

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

Capability update passed.

Command Result : No Error

lunacm:>hsm

4. Check that the new capability is in place:

lunacm:>hsm showpolicies

HSM Capabilities

```

0: Enable PIN-based authentication : 0
1: Enable PED-based authentication : 1
2: Performance level : 15
4: Enable domestic mechanisms & key sizes : 1
6: Enable masking : 0
7: Enable cloning : 1
8: Enable special cloning certificate : 0
9: Enable full (non-backup) functionality : 1
12: Enable non-FIPS algorithms : 1
15: Enable SO reset of partition PIN : 1
16: Enable network replication : 1
17: Enable Korean Algorithms : 1
18: FIPS evaluated : 0
19: Manufacturing Token : 0
20: Enable Remote Authentication : 1
21: Enable forcing user PIN change : 1
22: Enable offboard storage : 1
23: Enable partition groups : 0
25: Enable remote PED usage : 1
26: Enable External Storage of MTK Split : 0
27: HSM non-volatile storage space : 2097152
28: Enable HA mode CGX : 0
29: Enable Acceleration : 1
30: Enable unmasking : 1
31: Enable FW5 compatibility mode : 0
33: Maximum number of partitions : 100
34: Enable ECIES support : 0
35: Enable Single Domain : 1
36: Enable Unified PED Key : 1
37: Enable MofN : 1
38: Enable small form factor backup/restore : 0
39: Enable Secure Trusted Channel : 1
40: Enable decommission on tamper : 0
41: Enable Per-Partition SO : 1          <<=====
42: Enable partition re-initialize : 1

```

HSM Policies

```

0: PIN-based authentication : 0
1: PED-based authentication : 1
6: Allow masking : 0
7: Allow cloning : 1

```

```

12: Allow non-FIPS algorithms : 1
15: SO can reset partition PIN : 1
16: Allow network replication : 1
20: Allow Remote Authentication : 1
21: Force user PIN change after set/reset : 0
22: Allow offboard storage : 1
23: Allow partition groups : 0
25: Allow remote PED usage : 1
26: Store MTK Split Externally : 0
29: Allow Acceleration : 1
30: Allow unmasking : 1
31: Allow FW5 compatibility mode : 0
33: Current maximum number of partitions : 100
34: Allow ECIES support : 0
35: Force Single Domain : 0
36: Allow Unified PED Key : 0
37: Allow MofN : 1
38: Allow small form factor backup/restore : 0
39: Allow Secure Trusted Channel : 0
40: Allow decommission on tamper : 0
42: Allow partition re-initialize : 0

```

Command Result : No Error

lunacm:>

Serial Number Handling

Serial numbers of partitions changed after SafeNet HSM firmware 6.22.0.

The HSM serial number remains the HSM serial number, unchanged.

Application partition serial numbers are derived differently and are longer than they were previously.

Serial numbers for pre-existing application partitions (that were created on an HSM with firmware older than 6.22.0) are preserved when HSM firmware is updated.

New application partitions, created while an HSM is at firmware version 6.22.0 or newer, are assigned new, longer serial numbers, regardless of whether the partition is created as “legacy” or as PPSO type.

HA operation with SafeNet HSMs is unaffected by older serial numbers, new-style serial numbers, or any mix of the two. A failing member of an HA group can be replaced by a member with a different style of serial number.

If you have applications or scripts that rely on parsing HSM partition serial numbers (for example, log-file post-processing), some re-coding might be needed before you update.

Standards and Validations

When appropriate, we submit SafeNet products for validation/certification against international standards in the security, crypto, and data protection fields. This chapter discusses the major certifications that we routinely seek.



Note: If your application or your milieu does not require that HSM products carry validations like NIST's FIPS, or Common Criteria EAL, then this section would be of limited interest, except possibly for the reassurance that we design, test, manufacture, and handle our products

This chapter contains the following sections:

- "About FIPS Validation" below
- "About HSM NOT in FIPS140-2 Approved Mode" on the next page
- "NIST SP 800-131A: Changes to FIPS-Supported Algorithms Effective January 2014" on page 484
- "Common Criteria" on page 491

About FIPS Validation

In many areas of the information security industry, validations against independent or government standards are considered a desirable or essential attribute of a product. The pre-eminent standard in the field is the FIPS (Federal Information Processing Standards) 140 standard from the United States government's NIST (National Institute of Standards and Technology at <http://www.nist.gov>).

SafeNet routinely seeks FIPS validation for our HSM products. We were among the very first to achieve FIPS 140-1 validation for an HSM, and have since re-submitted our products when the standard changed (to 140-2) and whenever improvements to our product (or the introduction of new products) introduced enough change that the previous validation did not cover. Since the first, we have never failed to achieve the validation whenever we submit a product or variant.

In the case of appliance products, such as the SafeNet Network HSM validation is performed against the HSM Keycard inside the appliance, and not against the entire appliance. Furthermore, the validation of the HSM is for a particular firmware version, only.

However, the process of re-validation, though shorter than a new, first-time validation is nevertheless lengthy and involved. Therefore, whenever we introduce a new product, or a product variant that is sufficiently modified to require a new validation, there is a delay of several months until the validation certificate is granted.

Also to be considered is that validation can be a moving target. A product that received validation against the standard several years ago might not pass today, not because the standard has changed, but because interpretation has evolved or because testing organizations have revised their emphasis in some areas.

Finally, older standards give way to newer. Due to the expense and time constraints, companies tend to stay with a previous version of an optional standard until there is sufficient market demand for validation to the newer standard.

What does this mean to me?

Check with SafeNet to learn the most current validation status of any product, or click the NIST link, above. If FIPS validation is a primary concern for your application, you may need to use a previous, validated version of a product, and forego the latest improvements and features. During the lifetime of a product, we try to ensure that a FIPS-validated version remains available for purchase, despite the existence of newer firmware and hardware versions (see note below).

If the features of a new release are critical to your application, and FIPS validation is not a gating requirement, then you can use the newest product release. Based on past performance, the most recent release is likely to receive validation within months.



Note: It can happen that external circumstances make it impossible to fulfill the availability policy. An example from our own history was the introduction of the RoHS (Reduction of Hazardous Substances) legislation in the EU. Even where a company retained the capability to manufacture older-version units, they were no longer permitted to sell those older-design products to any country that abides by the RoHS regulations. In other cases, suppliers might no longer have stocked the non-RoHS parts that were used in the older design, and demand might not have justified their creating equivalent parts that meet RoHS standards. In such a situation, we could sell only the newer, RoHS-compliant product, while waiting for it to achieve FIPS (or other) validation.

"About HSM NOT in FIPS140-2 Approved Mode" below for more information.

About HSM NOT in FIPS140-2 Approved Mode

This is an option. You can change it.

If you run the `hsm show` command, you **might** see this text under the heading "FIPS 140-2 Operation":

The HSM is NOT in FIPS 140-2 approved operation mode

There is nothing wrong with your HSM. The message refers to an option (HSM policy code 12) that you can select according to your needs.

Here is how it works.

The FIPS-Approved Algorithms

The HSM is capable of a comprehensive set of cryptographic algorithms, allowing it to address the needs of a world data security market. However, some governments and agencies specify a restricted set of the available algorithms that suit their requirements or that limit the scope of testing that they are required to perform. FIPS is a very prominent set of standards in the industry, so we provide an option to exclude some algorithms from availability, so that an HSM owner can operate confidently in compliance with the FIPS 140-2 standard.

However, as technology advances and the cryptographic landscape shifts, agencies like NIST need to update their standards (like FIPS), dropping older, less effective options and adding newer ones as they become important and are vetted in the test labs. The list of FIPS-approved algorithms is subject to change.

For the most current list of FIPS approved algorithms, please visit the NIST web site at <http://csrc.nist.gov/>

For FIPS, you might be interested in:

- Modules in Process: <http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140InProcess.pdf>
- Completed Validations - Vendor List: <http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/1401vend.htm>

For Common Criteria EAL, you can check:

- Completed CC evaluations: <http://www.commoncriteriaportal.org/products/>

For Payment Card Industry standards:

- https://www.pcisecuritystandards.org/approved_companies_providers/approved_pin_transaction_security.php then click the "HSM" tab (the third item in the product category headings)

Because we make available more algorithms and mechanisms than are covered under FIPS testing and certification, we are required by the FIPS 140-2 standard to provide the **option** for FIPS approved algorithms **only**, and to warn users if they are in a mode of operation that allows non-FIPS approved algorithms.

See for summary table of our supported mechanisms, and their FIPS-approved status.

What Does This Mean For Your Application?

The only difference between the two modes is that when Non-FIPS algorithms are **disallowed**, then you are operating in fully FIPS-compliant mode, and you have access to **only** the algorithms listed by NIST for the standard (the algorithms that satisfy the standard). When Non-FIPS algorithms are allowed, then you have exactly the same HSM appliance, except that it now offers you the possibility to use many more algorithms that are not FIPS 140-2 approved. This is useful where FIPS 140-2 standard is not a requirement (in many countries and organizations around the world). That is, the HSM treats FIPS-allowed algorithms identically when in either mode, but in non-FIPS mode the HSM offers additional capability that is not available in FIPS mode.

Your choice, then, is which is more important in your situation:

- that you have the ability to use those additional algorithms with your application or,
- that you restrict yourself to using only the FIPS-approved algorithms (perhaps for policy or regulatory compliance)?

Thus, whoever is setting your policy must understand that it is the FIPS standard, and not SafeNet, that decides which algorithms are permissible. You then have the option to comply or not. The appliance remains just as highly secure in either mode, but you must choose the mode that satisfies your auditors and/or your application requirements.

What Are the Implications of Changing This Policy Setting?

Other than the compliance issue, described above, the major issue is that this is a destructive policy. That is, if you change this policy, the Partitions and their contents are lost. If you have any important keys, certificates, or other material stored on the SafeNet HSM, you will want to back them up before changing this, or any other destructive policy.

"About FIPS Validation" on page 459

Migrating from Non-FIPS HSM to FIPS HSM

If you have objects stored in an application partition of an HSM with FIPS mode off, you might encounter a situation where you need to begin using those objects on an HSM with FIPS mode in force. Depending on the HSM and other factors, you have some options.

One approach is to clone the needed objects to a different HSM that has the FIPS mode policy set on (or to archive (back up) the objects onto a Backup HSM and restore them onto the second HSM).

Another option, requiring fewer HSMs, is to change the policy on the current HSM from off to on. The policy change is destructive, meaning that all your objects would be destroyed. Therefore, you would need to archive (back up) the application partition contents before changing the HSM policy on the primary HSM, then create a new partition on that HSM, after the policy change, and restore the objects into that partition.

The exact sequence and commands to perform that operation might vary, depending on circumstances and on the firmware level of the HSM. As well, the commands are slightly different if you are using lunash on SafeNet Network HSM (for a legacy partition), or if you are using lunacm to perform the task on a SafeNet PCI-E HSM or on an NTLS-connected SafeNet Network HSM PPSO partition.



CAUTION: Before migrating to FIPS mode, check whether your Cryptoki application and the objects it uses for its crypto operations are suitable to run in FIPS mode for the particular version of SafeNet HSM firmware you are using. You can confirm that an application works in FIPS mode by setting up a test environment with a development HSM configured in FIPS mode. Specifically you will need to confirm that your application uses only FIPS approved mechanisms and key sizes. (See "Supported Mechanisms" on page 1.)

Be aware that FIPS object type and object size restrictions vary from firmware version to firmware version, usually becoming more restrictive in newer releases. If our documentation said that a particular mechanism or object type/size was acceptable in FIPS-mode for firmware that accompanied a previous release, that might no longer be the case if you are updating to a more current firmware version.

Start with objects in partition on Non-FIPS HSM	End with objects in partition on FIPS HSM	Lunash	Lunacm
SafeNet USB HSM	A different SafeNet USB HSM - (Backup HSM is not supported with SafeNet USB HSM)	n/a	- Clone the FIPS-approved objects individually from the HSM with FIPS policy OFF to the HSM with FIPS policy ON
SafeNet PCI-E HSM	A different SafeNet PCI-E HSM card in the same host	n/a	- Clone the FIPS-approved objects individually from the HSM with FIPS policy OFF to the HSM with FIPS policy ON
SafeNet PCI-E HSM	The same SafeNet PCI-E HSM card with the policy changed	n/a	- note the HSM and partition settings before proceeding, so you can replicate them after the change to FIPS mode - list the objects on the source partition

Start with objects in partition on Non-FIPS HSM	End with objects in partition on FIPS HSM	Lunash	Lunacm
			<ul style="list-style-type: none"> - for releases earlier than 6.0, review the "Mechanisms" chapter of the "SDK Reference Guide", to determine which (if any) objects on the source partition are not FIPS-approved. - for releases earlier than 6.0, delete from the source partition any objects that are not FIPS approved; you will no longer be able to store or use them when the HSM is in FIPS mode - Archive (back up) the objects from the primary HSM, with FIPS policy OFF, to a SafeNet Backup HSM - Switch the FIPS policy of the primary HSM to ON (which destroys the partition and all objects on the HSM) - re-initialize the HSM - create a new partition on the primary HSM - Restore all of the FIPS-acceptable objects from the backup HSM to the new partition on the HSM with FIPS policy ON - begin using FIPS-acceptable mechanisms to operate on/with the objects in the new partition
SafeNet Network HSM PPSO partition	A PPSO partition on the same SafeNet Network HSM with the policy changed	- note the HSM settings before proceeding, so you can replicate them after the change to FIPS mode [continue at client in lunacm] =====>>	<ul style="list-style-type: none"> - note the partition settings before proceeding, so you can replicate them after the change to FIPS mode - list the objects on the source partition - review the "Mechanisms" chapter of the "SDK Reference Guide", to determine which (if any) objects on the source partition are not FIPS-approved. - delete from the source partition any objects that are not FIPS approved; you will no longer be able to store or use them when the HSM is in FIPS mode

Start with objects in partition on Non-FIPS HSM	End with objects in partition on FIPS HSM	Lunash	Lunacm
		<ul style="list-style-type: none"> - Switch the FIPS policy of the primary HSM to ON (which destroys the partition and all objects on the HSM) - re-initialize the HSM - create a new partition on the primary HSM - restore client/partition assignments [conclude at client in lunacm] =====>> 	<ul style="list-style-type: none"> - Archive (back up) the objects from the source partition (on the HSM with FIPS policy OFF) to a SafeNet Backup HSM <<===== [continue at SafeNet Network HSM, in lunash] - Restore all of the FIPS-acceptable objects from the backup HSM to the new partition on the HSM with FIPS policy ON - begin using FIPS-acceptable mechanisms to operate on/with the objects in the new partition
SafeNet Network HSM legacy partition	A legacy partition on the same SafeNet Network HSM with the policy changed	<ul style="list-style-type: none"> - note the HSM and partition settings before proceeding, so you can replicate them after the change to FIPS mode - Archive (back up) the objects from the primary HSM with FIPS policy OFF to a SafeNet Backup HSM - Switch the FIPS policy of the primary HSM to ON (which destroys the partition and all objects on the HSM) - re-initialize the HSM - create a new partition on the primary HSM - Restore all of the FIPS-acceptable objects from the backup HSM to the new partition on the HSM with FIPS policy 	n/a

Start with objects in partition on Non-FIPS HSM	End with objects in partition on FIPS HSM	Lunash	Lunacm
		ON - restore client/partition assignments - begin using FIPS-acceptable mechanisms to operate on/with the objects in the new partition	



Note: After the HSM has been set to FIPS mode, only FIPS-approved objects, as allowed by the HSM's current version of firmware, can be restored onto the new partition.

For the KE (Key Export) HSM configuration, private keys cannot be backed up/restored. They can only be unwrapped back onto the new FIPS-mode partitions.

SafeNet Network HSM legacy application partition

The following instructions are for a legacy partition on a SafeNet Network HSM, because the legacy partition is administratively owned by the HSM SO, and all administrative commands are performed via LunaSH (lunash).

The following instructions assume that a SafeNet Backup HSM is connected locally to the SafeNet Network HSM appliance. If your SafeNet Backup HSM is located remotely from the SafeNet Network HSM, then you will need to modify these instructions by referring to additional information in ["Remote Application-Partition Backup and Restore Using the Backup HSM"](#) on page 68.

The following instructions assume that you have a number of objects on the application partition that you have been using while the HSM is non-FIPS, and that you wish to continue using after you switch the HSM to FIPS mode.

For illustrative purposes, the example source partition in the following instructions includes a few objects that are not permitted in an HSM running in FIPS mode.

1. View the HSM information for the SafeNet HSM that contains the source partition. Confirm that the HSM is "...NOT in FIPS 140-2 approved operation mode" before the HSM is switched to FIPS mode.

```
[192.20.9.127] lunash:>hsm show
Appliance Details:
=====
Software Version:          6.0.0-39

HSM Details:
=====
HSM Label:                 JR
Serial #:                  384
Firmware:                  6.10.6
HSM Model:                 K6Base
Authentication Method:     PED keys
```

```

HSM Admin login status:      Logged In
HSM Admin login attempts left: 3 before HSM zeroization!
RPV Initialized:             Yes
Audit Role Initialized:      Yes
Remote Login Initialized:     No
Manually Zeroized:           No

```

Partitions created on HSM:

```

=====
Partition:      894069505, Name: mylegacypar1

```

```

Number of partitions allowed: 20
Number of partitions created: 1

```

FIPS 140-2 Operation:

```

=====
The HSM is NOT in FIPS 140-2 approved operation mode.

```

HSM Storage Information:

```

=====
Maximum HSM Storage Space (Bytes): 2097152
Space In Use (Bytes):              104857
Free Space Left (Bytes):           1992295

```

```

Command Result : 0 (Success)
[192.20.9.127] lunash:>

```

2. View the the policy settings of the non-FIPS HSM partition that is to be backed up, so that you can reproduce the settings in the newly-created partition after switching to FIPS mode.

```

[192.20.9.127] lunash:>partition showpolicies -partition mylegacypar1

```

```

Partition Name:      mylegacypar1
Partition SN:        384013
Partition Label:     MyLegacyPar1
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

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	On	22
Allow auto-activation	On	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

Command Result : 0 (Success)
[172.20.9.127] lunash:>

3. Use **partition showcontents** command to view the contents of the non-FIPS HSM partition to be backed up.
Type:

```
[192.20.9.127] lunash:>partition showcontents -partition mylegacypar1 -password userpin
```

```
Partition Name:          mylegacypar1
Partition SN:            894069505
Partition Label:         MyLegacyPar1
```

```
Storage (Bytes): Total=102701, Used=2436, Free=100265
Number objects: 7
```

```
Object Label:  Generated AES Key
Object Type:   Symmetric Key
Object Handle: 20
```

```
Object Label:  X9_62_prime256v1(P-256)+ ECDSA Public Key
Object Type:   Public Key
Object Handle: 21
```

```
Object Label:  RSA 186-3 Aux Primes 2048bit Private Key
Object Type:   Private Key
Object Handle: 28
```

```
Object Label:  Generated DES3 Key
Object Type:   Symmetric Key
Object Handle: 31
```

```
Object Label:  X9_62_prime256v1(P-256)+ ECDSA Private Key
Object Type:   Private Key
Object Handle: 32
```

```
Object Label:  RSA 186-3 Aux Primes 2048bit Public Key
Object Type:   Public Key
Object Handle: 33
```

```
Object Label:  User Private RSA Key5-1024
Object Type:   Private Key
Object Handle: 66
```

```
Object Label:  User Public RSA Key5-1024
Object Type:   Public Key
Object Handle: 65
```

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

4. Check the information of the Backup HSM, as you will need to use its serial number in the backup command:

```
[192.20.9.127] lunash:>token backup list
```

```
Token Details:
=====
Token Label:          G5Backupup
Slot:                 5
Serial #:              475292
```



```
Firmware:                6.0.8
HSM Model:               G5Backup
```

Command Result : 0 (Success)

[192.20.9.127] lunash:>

5. Backup the contents of the non-FIPS partition to the SafeNet Backup HSM:

```
[192.20.9.127] lunash:>partition backup -partition mylegacypar1 -password userpin -serial
475292 -tokenpar mypartitionbackup1
```

Warning: You will need to connect Luna PED to the SafeNet Backup HSM to complete this operation.

You may use the same Luna PED that you used for SafeNet Network HSM.

Please type 'proceed' and hit <enter> when you are ready to proceed> proceed

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

Luna PED operation required to login to user on token - use User or Partition Owner (black) PED key.

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

Please enter the password for the HSM user partition:

> *****

Warning: You will need to connect Luna PED to the SafeNet Backup HSM to complete this operation.

You may use the same Luna PED that you used for SafeNet Network HSM.

Please type 'proceed' and hit <enter> when you are ready to proceed> proceed

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

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

Luna PED operation required to login to user on token - use User or Partition Owner (black) PED key.

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

```
Object "Generated AES Key" (handle 20) cloned to handle 242 on target
Object "X9_62_prime256v1(P-256)+ ECDSA Public Key" (handle 21) cloned to handle 243 on target
Object "RSA 186-3 Aux Primes 2048bit Private Key" (handle 28) cloned to handle 244 on target
Object "Generated DES3 Key" (handle 31) cloned to handle 247 on target
Object "X9_62_prime256v1(P-256)+ ECDSA Private Key" (handle 32) cloned to handle 248 on target
Object "RSA 186-3 Aux Primes 2048bit Public Key" (handle 33) cloned to handle 249 on target
Object "User Private RSA Key5-1024" (handle 66) cloned to handle 87 on target
Object "User Public RSA Key5-1024" (handle 65) cloned to handle 156 on target
'partition backup' successful.
```

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

6. Log into the HSM as SO, so that you can change an HSM policy:

```
[172.20.9.127] lunash:>hsm login
```

Luna PED operation required to login as HSM Administrator - use Security Officer (blue) PED key.

```
'hsm login' successful.
```

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

7. Verify the HSM policies before performing the change:

```
[192.20.9.127] lunash:>hsm showpolicies
```

```
HSM Label:   JR
Serial #:    384
Firmware:    6.10.6
```

The following capabilities describe this HSM, and cannot be altered except via firmware or capability updates.

Description	Value
=====	=====
Enable PIN-based authentication	Disallowed
Enable PED-based authentication	Allowed
Performance level	15
Enable domestic mechanisms & key sizes	Allowed
Enable masking	Allowed
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	Disallowed
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	Allowed
Enable External Storage of MTK Split	Allowed
HSM non-volatile storage space	2097152
Enable HA mode CGX	Disallowed
Enable Acceleration	Allowed
Enable unmasking	Disallowed
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
=====	=====
PED-based authentication	True
Store MTK Split Externally	False

The following policies describe the current configuration of this HSM and may be changed by the HSM Administrator.

Changing policies marked "destructive" will zeroize (erase completely) the entire HSM.

Description	Value	Code	Destructive
=====	=====	=====	=====
Allow masking	On	6	Yes
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 remote PED usage	On	25	No
Allow Acceleration	On	29	Yes

Command Result : 0 (Success)

[192.20.9.127] lunash:>

8. Change (destructive) HSM Policy 12 from 1 to 0, setting FIPS mode ON for the HSM:

[192.20.9.127] lunash:>hsm changepolicy -policy 12 -value 0

CAUTION: Are you sure you wish to change the destructive policy named:

Allow non-FIPS algorithms

Changing this policy will result in erasing all partitions on the HSM! (HSM Admin, Domain, and M of N (where applicable) will not be modified.)

Type 'proceed' to zeroize your HSM and change the policy, or 'quit' to quit now.

> proceed

'hsm changePolicy' successful.

Policy Allow non-FIPS algorithms is now set to value: 0

Command Result : 0 (Success)

[192.20.9.127] lunash:>

9. Verify that the FIPS 140-2 mode is now ON for the HSM:

[192.20.9.127] lunash:>hsm showpolicies

HSM Label: JR

Serial #: 384
Firmware: 6.10.6

The following capabilities describe this HSM, and cannot be altered except via firmware or capability updates.

Description	Value
=====	=====
Enable PIN-based authentication	Disallowed
Enable PED-based authentication	Allowed
Performance level	15
Enable domestic mechanisms & key sizes	Allowed
Enable masking	Allowed
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	Disallowed
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	Allowed
Enable External Storage of MTK Split	Allowed
HSM non-volatile storage space	2097152
Enable HA mode CGX	Disallowed
Enable Acceleration	Allowed
Enable unmasking	Disallowed
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
=====	=====
PED-based authentication	True
Store MTK Split Externally	False

The following policies describe the current configuration of this HSM and may be changed by the HSM Administrator.

Changing policies marked "destructive" will zeroize (erase completely) the entire HSM.

Description	Value	Code	Destructive
=====	=====	=====	=====
Allow masking	On	6	Yes
Allow cloning	On	7	Yes

Allow non-FIPS algorithms	Off	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 remote PED usage	On	25	No
Allow Acceleration	On	29	Yes

Command Result : 0 (Success)
[192.20.9.127] lunash:>

10. Log into the HSM:

```
[192.20.9.127] lunash:>hsm login
Luna PED operation required to login as HSM Administrator - use Security Officer (blue) PED
key.
'hsm login' successful.
Command Result : 0 (Success)
[192.20.9.127] lunash:>
```

11. Create a partition to which [some of] the backed-up objects are to be restored:



Note: When creating the new application partition to receive the objects that were archived, you **MUST** use the same domain as was associated with the original (source) application partition (the same red PED Key for PED-authenticated HSM, or the same domain string for password-authenticated HSM).

You should also use the same partition name, if your application expects to find the partition by that name.

```
[192.20.9.127] lunash:>partition create -partition mylegacypar1 -label MyLegacyPar1
On completion, you will have this number of partitions: 1
```

```
Type 'proceed' to create the initialized partition, or
'quit' to quit now.
> proceed
```

Please ensure that you copy the password from the Luna PED and that you keep it in a safe place.

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

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

'partition create' successful.

Command Result : 0 (Success)
[192.20.9.127] lunash:>

12. Re-set the partition policies to previous values and change the partition challenge, if desired, at this time:

```
[192.20.9.127] lunash:>partition showpolicies -partition mylegacyparl
```

```
Partition Name:          mylegacyparl
Partition SN:            384013
Partition Label:         MyLegacyParl
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

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

Command Result : 0 (Success)

[192.20.9.127] lunash:>

Compare the output with the policy settings before the original partition was destroyed, and make any changes necessary.

13. Restore the contents from the Backup HSM to the new application partition on the now-FIPS-mode HSM:

```
[192.20.9.127] lunash:>partition restore -partition mylegacypar1 -tokenPar mypar-
titionbackup1 -serial 475292 -add
```

Warning: You will need to connect Luna PED to the SafeNet Backup HSM to complete this operation.

You may use the same Luna PED that you used for SafeNet Network HSM.

Please type 'proceed' and hit <enter> when you are ready to proceed> proceed

Luna PED operation required to login to user on token - use User or Partition Owner (black) PED key.

```
Object "Generated AES Key" (handle 242) cloned to handle 20 on target
Object "X9_62_prime256v1(P-256)+ ECDSA Public Key" (handle 243) cloned to handle 21 on target
Object "RSA 186-3 Aux Primes 2048bit Private Key" (handle 244) cloned to handle 30 on target
Object "Generated DES3 Key" (handle 247) cloned to handle 33 on target
Object "X9_62_prime256v1(P-256)+ ECDSA Private Key" (handle 248) cloned to handle 34 on tar-
get
Object "RSA 186-3 Aux Primes 2048bit Public Key" (handle 249) cloned to handle 35 on target
Object "User Private RSA Key5-1024" (handle 87) cloned to handle 81 on target
Error: 'partition restore' failed. (130000 : LUNA_RET_ATTRIBUTE_VALUE_INVALID)
Object "User Public RSA Key5-1024" (handle 156) cloned to handle 140 on target
Error: 'partition restore' failed. (130000 : LUNA_RET_ATTRIBUTE_VALUE_INVALID)
```

Command Result : 65535 (Luna Shell execution)

[192.20.9.127] lunash:>



Note: In this example, the restore operation actually passed, but you see an error message because one of the objects was not FIPS-valid, and therefore could not be accepted by the partition on the (now) FIPS-mode HSM. Any invalid objects are skipped, and all valid objects are restored onto the new partition.

14. Verify the objects that have been successfully restored onto the new application partition.

```
[192.20.9.127] lunash:>partition showcontents -partition mylegacypar1
Please enter the password for the partition:
> *****
```

```
Partition Name:                mylegacyfipspar1
Partition SN:                  894069506
Partition Label:               MyLegacyFIPSPar1
Storage (Bytes): Total=95569, Used=37760, Free=57809
Number objects: 6
```

```
Object Label:  Generated AES Key
Object Type:   Symmetric Key
Object Handle: 20
```

```
Object Label:  X9_62_prime256v1(P-256)+ ECDSA Public Key
Object Type:   Public Key
Object Handle: 21
```

```
Object Label:  RSA 186-3 Aux Primes 2048bit Private Key
Object Type:   Private Key
Object Handle: 28
```

```
Object Label:  Generated DES3 Key
Object Type:   Symmetric Key
Object Handle: 31
```

```
Object Label:  X9_62_prime256v1(P-256)+ ECDSA Private Key
Object Type:   Private Key
Object Handle: 32
```

```
Object Label:  RSA 186-3 Aux Primes 2048bit Public Key
Object Type:   Public Key
Object Handle: 33
```

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

All the uncontested objects (those that are acceptable in FIPS 140-2 mode) have been restored onto the new application partition on your HSM. The objects for which an error was reported, meaning that they are not allowed in FIPS mode, could not be restored to the partition now that the HSM is in FIPS 140-2 mode. You will need to find other ways to accomplish whatever you were doing with those excluded objects, now that your operation is constrained to FIPS 140-2 mode.

SafeNet Network HSM or SafeNet PCI-E HSM application partition (f/w 6.22.0 or newer) in LunaCM

The following instructions are for

- a SafeNet PCI-E HSM application partition (locally installed), or
- a SafeNet Network HSM application partition via secure network connection (NTLS or STC)

In both cases, the HSM firmware is version 6.22.0 or newer. The SafeNet Network HSM application partition is a PPSO partition (not legacy), which can be administered only via client connection (lunacm).

The following instructions assume that a SafeNet Backup HSM is connected locally to the SafeNet HSM Client host computer.

The following instructions assume that you have a number of objects on the application partition that you have been using while the HSM is non-FIPS, and that you wish to continue using after you switch the HSM to FIPS mode.

For illustrative purposes, the example source partition in the following instructions includes a few objects that are not permitted in an HSM running in FIPS mode. This includes verifying the HSM's FIPS mode,

For SafeNet Network HSM, the initial action to set up the application partition must be performed by the HSM SO at the SafeNet Network HSM appliance using LunaSH, via SSH or via local serial console.

For SafeNet PCI-E HSM, the HSM SO creates the partition in lunacm.

Thereafter, either for a SafeNet PCI-E HSM local application partition (with PSO), or for a SafeNet Network HSM application partition (with PSO) viewed over NTLS connection, the actions are the same and are performed in lunacm as follows.

1. Use **slot set slot** command to set the current-slot focus to the source application partition on the non-FIPS-mode HSM. Type:

```
lunacm:> slot set slot 1

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

Command Result : No Error

lunacm:>
```

2. Use **partition showinfo** command to view the HSM information for the SafeNet HSM that contains the source partition. Confirm that the HSM is "...NOT in FIPS 140-2 approved operation mode" before the HSM is switched to FIPS mode.

```
lunacm:> partition showinfo

Partition Label -> Cryptoki User
Partition Manufacturer -> Safenet, Inc.
Partition Model -> K6 Base
Partition Serial Number -> 450039014
Partition Status -> OK
Token Flags ->
    CKF_LOGIN_REQUIRED
    CKF_USER_PIN_INITIALIZED
    CKF_RESTORE_KEY_NOT_NEEDED
    CKF_PROTECTED_AUTHENTICATION_PATH
    CKF_TOKEN_INITIALIZED
RPV Initialized -> Yes
```

```

Slot Id -> 1
Tunnel Slot Id -> 4
Session State -> CKS_RW_USER_FUNCTIONS
Role Status -> Crypto Officer logged in
Token Flags ->
    TOKEN_KCV_CREATED
Partition OID: 40000006f3040000f7dd0600

```

```

Partition Storage:
    Total Storage Space: 2087864
    Used Storage Space: 0
    Free Storage Space: 2087864
    Object Count: 0
    Overhead: 9288

```

*** The partition is in FIPS 140-2 approved operation mode. ***

Command Result : No Error

```
lunacm:> slot set slot 6
```

```
Current Slot Id: 6 (SafeNet USB HSM 6.0.8 (PED) Backup Device)
```

Command Result : No Error

```
lunacm:> hsm si
```

```

HSM Label -> G5Backup
HSM Manufacturer -> Safenet, Inc.
HSM Model -> G5Backup
HSM Serial Number -> 475292
HSM Status -> OK
Token Flags ->
    CKF_RNG
    CKF_LOGIN_REQUIRED
    CKF_USER_PIN_INITIALIZED
    CKF_RESTORE_KEY_NOT_NEEDED
    CKF_PROTECTED_AUTHENTICATION_PATH
    CKF_TOKEN_INITIALIZED
Firmware Version -> 6.0.8
Rollback Firmware Version -> Not Available

```

```

HSM Auditor: Not Supported
HSM Logging: Not Supported

```

```

RPV Initialized -> No
Slot Id -> 6
Session State -> CKS_RW_PUBLIC_SESSION

```

```
SO Status-> Not Logged In
```

```

SO Failed Logins-> 0
SO Flags ->
    CONTAINER_KCV_CREATED

```

```
HSM Storage:
```

```

Total Storage Space: 16252928
Used Storage Space: 147352
Free Storage Space: 16105576
Allowed Partitions: 20
Number of Partitions: 7

```

SO Storage:

```

Total Storage Space: 262144
Used Storage Space: 0
Free Storage Space: 262144
Object Count: 0

```

*** The HSM is NOT in FIPS 140-2 approved operation mode. ***

License Count -> 5

```

1. 621010355-000 SafeNet Remote Backup HSM base configuration
1. 621000006-001 Enabled for 15.5 megabytes of object storage
1. 621000007-001 Enable the master tamper key to be stored externally
1. 621000008-001 Enable remote PED capability
1. 621000005-001 Maximum 20 partitions

```

Command Result : No Error

lunacm:>

3. Use **slot set -slot** command to set the current-slot focus to the slot with the Backup HSM. Type:

lunacm:> slot set -s 6

```

Current Slot Id:      6      (SafeNet USB HSM 6.0.8 (PED) Backup Device)

```

Command Result : No Error

lunacm:>

4. Use **partition archive list** to show the contents of the backup partition. Type:

lunacm:> partition archive list -slot 6

HSM Storage Information for slot 6:

```

Total HSM Storage Space: 16252928
Used HSM Storage Space: 147352
Free HSM Storage Space: 16105576
Allowed Partitions: 20
Number Of Partitions: 7

```

Partition list for slot 6

Number of partition: 7

```

Name:                John2Backup2
Total Storage Size: 1684
Used Storage Size: 1643
Free Storage Size: 41
Number Of Objects: 2

```

```

Name: JohnK6backup
Total Storage Size: 41340
Used Storage Size: 40210
Free Storage Size: 1130
Number Of Objects: 85

Name: 6106Backup
Total Storage Size: 41340
Used Storage Size: 40210
Free Storage Size: 1130
Number Of Objects: 85

Name: UserPartitionBackup
Total Storage Size: 2436
Used Storage Size: 2353
Free Storage Size: 83
Number Of Objects: 6

Name: UserPartitionBackup1
Total Storage Size: 2436
Used Storage Size: 2353
Free Storage Size: 83
Number Of Objects: 6

Name: John2Backup
Total Storage Size: 1684
Used Storage Size: 1643
Free Storage Size: 41
Number Of Objects: 2

Name: 6101Backup
Total Storage Size: 41340
Used Storage Size: 40210
Free Storage Size: 1130
Number Of Objects: 85

```

Command Result : No Error

lunacm:>

5. Use **hsm showinfo** to verify that the newly initialized HSM is now in FIPS mode. Type:

lunacm:> hsm showinfo

```

HSM Label -> G5Backkup
HSM Manufacturer -> Safenet, Inc.
HSM Model -> G5Backup
HSM Serial Number -> 475292
HSM Status -> OK
Token Flags ->
    CKF_RNG
    CKF_LOGIN_REQUIRED
    CKF_USER_PIN_INITIALIZED
    CKF_RESTORE_KEY_NOT_NEEDED
    CKF_PROTECTED_AUTHENTICATION_PATH
    CKF_TOKEN_INITIALIZED

```

```

Firmware Version -> 6.0.8
Rollback Firmware Version -> Not Available

HSM Auditor: Not Supported
HSM Logging: Not Supported

RPV Initialized -> No
Slot Id -> 6
Session State -> CKS_RW_PUBLIC_SESSION

SO Status-> Not Logged In

SO Failed Logins-> 0
SO Flags ->
    CONTAINER_KCV_CREATED

HSM Storage:
    Total Storage Space: 16252928
    Used Storage Space: 147352
    Free Storage Space: 16105576
    Allowed Partitions: 20
    Number of Partitions: 7

SO Storage:
    Total Storage Space: 262144
    Used Storage Space: 0
    Free Storage Space: 262144
    Object Count: 0

*** The HSM is NOT in FIPS 140-2 approved operation mode. ***

License Count -> 5
    1. 621010355-000 SafeNet Remote Backup HSM base configuration
    1. 621000006-001 Enabled for 15.5 megabytes of object storage
    1. 621000007-001 Enable the master tamper key to be stored externally
    1. 621000008-001 Enable remote PED capability
    1. 621000005-001 Maximum 20 partitions

Command Result : No Error

lunacm:> slot set -s 1

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

Command Result : No Error

lunacm:> role show -name Crypto Officer

    State of role 'Crypto Officer':
        Primary authentication type: PED
        Secondary authentication type: PIN
        Failed login attempts before lockout: 10

Command Result : No Error

lunacm:>

```

6. Use **partition showinfo** to verify that the new partition is in FIPS mode. Type:

```
lunacm:> partition showinfo

Partition Label -> Cryptoki User
Partition Manufacturer -> Safenet, Inc.
Partition Model -> K6 Base
Partition Serial Number -> 450039014
Partition Status -> OK
Token Flags ->
    CKF_LOGIN_REQUIRED
    CKF_USER_PIN_INITIALIZED
    CKF_RESTORE_KEY_NOT_NEEDED
    CKF_PROTECTED_AUTHENTICATION_PATH
    CKF_TOKEN_INITIALIZED
RPV Initialized -> Yes
Slot Id -> 1
Tunnel Slot Id -> 4
Session State -> CKS_RW_USER_FUNCTIONS
Role Status -> Crypto Officer logged in
Token Flags ->
    TOKEN_KCV_CREATED
Partition OID: 40000006f3040000f7dd0600

Partition Storage:
    Total Storage Space: 2087864
    Used Storage Space: 0
    Free Storage Space: 2087864
    Object Count: 0
    Overhead: 9288

*** The partition is in FIPS 140-2 approved operation mode. ***

Command Result : No Error

lunacm:>

lunacm:>
```

7. Use **partition archive restore** to restore the contents of the backup partition onto the newly created application partition on the now FIPS-mode HSM. Type:

```
lunacm:> partition archive restore -slot 6 -partition 6106Backup

Logging in to partition 6106Backup on slot 6 as the user.

Please attend to the PED.

Verifying that all objects can be restored...

85 objects will be restored.

Restoring objects...
Cloned object 122 from partition 6106Backup (new handle 20).
```

Cloned object 123 from partition 6106Backup (new handle 25).
Cloned object 124 from partition 6106Backup (new handle 26).
Cloned object 125 from partition 6106Backup (new handle 27).
Cloned object 126 from partition 6106Backup (new handle 28).
Cloned object 127 from partition 6106Backup (new handle 29).
Failed to clone object 128 from partition 6106Backup (CKR_ATTRIBUTE_VALUE_INVALID).
Cloned object 129 from partition 6106Backup (new handle 30).
Failed to clone object 130 from partition 6106Backup (CKR_ATTRIBUTE_VALUE_INVALID).
Failed to clone object 131 from partition 6106Backup (CKR_KEY_TYPE_INCONSISTENT).
Cloned object 132 from partition 6106Backup (new handle 31).
Cloned object 133 from partition 6106Backup (new handle 32).
Cloned object 136 from partition 6106Backup (new handle 35).
Cloned object 137 from partition 6106Backup (new handle 36).
Cloned object 138 from partition 6106Backup (new handle 37).
Cloned object 139 from partition 6106Backup (new handle 38).
Cloned object 143 from partition 6106Backup (new handle 42).
Cloned object 144 from partition 6106Backup (new handle 43).
Cloned object 145 from partition 6106Backup (new handle 44).
Cloned object 148 from partition 6106Backup (new handle 47).
Cloned object 149 from partition 6106Backup (new handle 48).
Cloned object 150 from partition 6106Backup (new handle 49).
Cloned object 151 from partition 6106Backup (new handle 50).
Cloned object 152 from partition 6106Backup (new handle 51).
Failed to clone object 153 from partition 6106Backup (CKR_KEY_TYPE_INCONSISTENT).
Cloned object 154 from partition 6106Backup (new handle 52).
Cloned object 158 from partition 6106Backup (new handle 56).
Cloned object 159 from partition 6106Backup (new handle 57).
Cloned object 160 from partition 6106Backup (new handle 58).
Cloned object 161 from partition 6106Backup (new handle 59).
Failed to clone object 162 from partition 6106Backup (CKR_ATTRIBUTE_VALUE_INVALID).
Cloned object 163 from partition 6106Backup (new handle 60).
Cloned object 164 from partition 6106Backup (new handle 61).
Cloned object 165 from partition 6106Backup (new handle 62).
Failed to clone object 166 from partition 6106Backup (CKR_ATTRIBUTE_VALUE_INVALID).
Cloned object 167 from partition 6106Backup (new handle 63).
Cloned object 168 from partition 6106Backup (new handle 64).
Cloned object 169 from partition 6106Backup (new handle 65).
Cloned object 170 from partition 6106Backup (new handle 66).
Cloned object 171 from partition 6106Backup (new handle 67).
Cloned object 175 from partition 6106Backup (new handle 71).
Failed to clone object 176 from partition 6106Backup (CKR_KEY_TYPE_INCONSISTENT).
Cloned object 177 from partition 6106Backup (new handle 72).
Cloned object 180 from partition 6106Backup (new handle 75).
Cloned object 181 from partition 6106Backup (new handle 76).
Cloned object 182 from partition 6106Backup (new handle 77).
Cloned object 183 from partition 6106Backup (new handle 78).
Failed to clone object 184 from partition 6106Backup (CKR_ATTRIBUTE_VALUE_INVALID).
Cloned object 185 from partition 6106Backup (new handle 79).
Cloned object 186 from partition 6106Backup (new handle 80).
Cloned object 187 from partition 6106Backup (new handle 81).
Cloned object 188 from partition 6106Backup (new handle 82).
Cloned object 189 from partition 6106Backup (new handle 83).
Failed to clone object 190 from partition 6106Backup (CKR_ATTRIBUTE_VALUE_INVALID).
Failed to clone object 191 from partition 6106Backup (CKR_ATTRIBUTE_VALUE_INVALID).
Cloned object 192 from partition 6106Backup (new handle 84).
Cloned object 193 from partition 6106Backup (new handle 85).
Cloned object 194 from partition 6106Backup (new handle 86).

```

Cloned object 198 from partition 6106Backup (new handle 90).
Cloned object 199 from partition 6106Backup (new handle 91).
Cloned object 200 from partition 6106Backup (new handle 92).
Cloned object 201 from partition 6106Backup (new handle 93).
Cloned object 204 from partition 6106Backup (new handle 96).
Cloned object 205 from partition 6106Backup (new handle 97).
Cloned object 206 from partition 6106Backup (new handle 98).
Cloned object 207 from partition 6106Backup (new handle 99).
Cloned object 208 from partition 6106Backup (new handle 100).
Cloned object 209 from partition 6106Backup (new handle 101).
Cloned object 210 from partition 6106Backup (new handle 102).
Failed to clone object 211 from partition 6106Backup (CKR_KEY_TYPE_INCONSISTENT).
Failed to clone object 212 from partition 6106Backup (CKR_ATTRIBUTE_VALUE_INVALID).
Failed to clone object 213 from partition 6106Backup (CKR_ATTRIBUTE_VALUE_INVALID).
Cloned object 214 from partition 6106Backup (new handle 103).
Cloned object 215 from partition 6106Backup (new handle 104).
Cloned object 216 from partition 6106Backup (new handle 105).
Cloned object 217 from partition 6106Backup (new handle 106).
Failed to clone object 218 from partition 6106Backup (CKR_ATTRIBUTE_VALUE_INVALID).
Cloned object 219 from partition 6106Backup (new handle 107).
Cloned object 220 from partition 6106Backup (new handle 108).
Cloned object 221 from partition 6106Backup (new handle 109).
Cloned object 222 from partition 6106Backup (new handle 110).
Cloned object 223 from partition 6106Backup (new handle 111).
Cloned object 227 from partition 6106Backup (new handle 115).
Failed to clone object 228 from partition 6106Backup (CKR_KEY_TYPE_INCONSISTENT).
Cloned object 229 from partition 6106Backup (new handle 116).

```

Not all objects can be cloned. Please verify HSM configuration.

Restore Complete.

70 objects have been restored from partition 6106Backup on slot 6.

Command Result : No Error

lunacm:>

NIST SP 800-131A: Changes to FIPS-Supported Algorithms Effective January 2014

As a result of the NIST SP 800-131A algorithm transition, the list of algorithms supported in FIPS mode is changing. These changes come into effect on 01 January 2014 .

Summary

To comply with this change, the following algorithms are not supported in SafeNet HSM 5.4, and higher, when the HSM is operated in FIPS mode:

- All digital signature and mac generation algorithms that use SHA-1 will no longer be supported, digital signature verification and mac verification will still be supported using SHA-1 for legacy purposes
- DSA Key Pair Generation and Signature Generation with a key size of less than 2048 bits is no longer supported
- DSA Signature Verification of 1024 bit keys is still supported for legacy purposes
- RSA Key Pair Generation and Signature Generation with a key size of less than 2048 bits is no longer supported
- RSA Signature Verification of 1024 bit keys is still supported for legacy purposes
- ECDSA DSA Key Pair Generation and Signature Generation with a curve size of less than 224 bits is no longer supported
- ECDSA Signature Verification with a curve size of less than 224 is still supported for legacy purposes
- RSA Key wrapping with an RSA Key of less than 2048 bits is no longer supported, however key unwrapping is still supported for legacy purposes
- RSA encryption with an RSA key of less than 2048 bits is no longer supported, however decryption is still supported for legacy purposes
- Diffie-Hellman key agreement with a key size of less than 2048 bits is no longer supported
- EC Diffie-Hellman key agreement with a curve size of less than 224 bits is no longer supported
- HMAC Generation with a key size less than 112 bits is no longer supported
- HMAC Verification with a key size less than 112 bits is supported for legacy purposes



Note: Use of SHA-1 is allowed for use in FIPS Approved mode, with the exception of digital signature/ MAC generation applications, for which it is not allowed in FIPS Mode.

Affected Algorithms

These changes affect the following algorithms:

Digital Signature Changes

Digital Signature	Key Pair Generation	Signature Generation	Signature Verification
DSA < 2048 with SHA-1	OFF	OFF	LEGACY
DSA < 2048 with SHA-2	OFF	OFF	LEGACY
RSA < 2048 with SHA-1	OFF	OFF	LEGACY
RSA < 2048 with SHA-2	OFF	OFF	LEGACY
ECDSA n < 224 with SHA-1	OFF	OFF	LEGACY
ECDSA n < 224 with SHA-2	OFF	OFF	LEGACY

Key Transport Changes

	Key Wrapping	Key Unwrapping
RSA < 2048	OFF	LEGACY

Encryption Changes

	Encryption	Decryption
RSA < 2048	OFF	LEGACY

Key Agreement Changes

	Key Agreement
Diffie-Hellman < 2048	OFF
EC Diffie-Hellman with n < 224	OFF

2-Key Triple DES Changes

	Encryption	Decryption	Key Wrapping	Key Unwrapping	CMAC KDF	HMAC KDF	CMAC Generation	CMAC Verification
2-Key Triple DES	RESTRICTED	LEGACY	RESTRICTED	LEGACY	DEPRECATED	ACCEPTABLE	DEPRECATED	LEGACY



Note: Restricted key types must be enforced at the application level.

NIST SP 800-131A restriction implementation in your application

For practical and performance reasons, we do not implement these FIPS restrictions in firmware; therefore if you wish to use 2-Key Triple DES in FIPS compliant manner, then you must implement the needed restrictions within your application, as follows.

As specified in the SP 800-131A document: “the use of the algorithm or key length is deprecated, and there are additional restrictions required to use the algorithm or key length for applying cryptographic protection to data”.

2-Key Triple DES Encryption:

The use of two-key Triple DES is acceptable for encryption through December 31, 2010.

From January 1, 2011 through December 31, 2015, the use of two-key Triple DES for encryption is **restricted**: the total number of blocks of data encrypted with the same cryptographic key **shall not** be greater than 2^{20} (note that for this algorithm, a block is the 64-bit block of a Triple DES encryption operation). This restriction also applies to those keys that were first used prior to 2011 and continue to be used beyond December 31, 2010 (i.e., those keys whose

cryptoperiod begins prior to 2011 and extends into 2011). Rationale for this exception is provided in Appendix A.1. After December 31, 2015, two-key Triple DES **shall not** be used for encryption.

2-Key Triple DES Wrapping:

Two-key Triple DES is acceptable for wrapping and unwrapping keying material through December 31, 2010.

From January 1, 2011 through December 31, 2015, the use of two-key Triple DES for wrapping keying material is **restricted**: the total number of blocks of keying material wrapped with the same cryptographic key **shall** be no more than 2^{20} .

After December 31, 2015, two-key Triple DES **shall not** be used to wrap keying material.

HMAC Changes

	MAC Generation	MAC Verification
HMAC < 112	OFF	LEGACY



Note: SHA-1 is allowed except for digital signature/MAC Generation

Impact on your operations

You can restore keys having legacy bit lengths from a backup. Legacy keys are retained on the HSM after the upgrade to SafeNet HSM 5.4 or later, and function in 'legacy' mode, only.

If you still wish to use the 'legacy' keys fully, you must exit FIPS mode:

- Backup your keys
- Switch off FIPS mode (change the policy), wiping out all keys
- Restore keys to the HSM that is no longer in FIPS mode

Mechanisms Affected

These changes affect the following mechanisms:

RSA FIPS Mechanisms

RSA FIPS Mechanism	FIPS	Changes in FIPS mode
CKM_RSA_PKCS_KEY_PAIR_GEN	YES	LEGACY less than 2048 bit
CKM_RSA_PKCS	YES	LEGACY less than 2048 bit
CKM_SHA1_RSA_PKCS	YES	LEGACY
CKM_RSA_PKCS_OAEP	YES	LEGACY less than 2048 bit
CKM_RSA_X9_31_KEY_PAIR_GEN	YES	LEGACY less than 2048 bit
CKM_RSA_FIPS_186_3_AUX_PRIME_KEY_PAIR_GEN	YES	LEGACY less than 2048 bit
CKM_RSA_FIPS_186_3_PRIME_KEY_PAIR	YES	NO, Already enforced at 2048 bit

RSA FIPS Mechanism	FIPS	Changes in FIPS mode
CKM_RSA_X9_31_KEY_PAIR_GEN	YES	LEGACY less than 2048 bit
CKM_SHA1_RSA_X9_31	YES	LEGACY
CKM_SHA224_RSA_X9_31	YES	LEGACY less than 2048 bit
CKM_SHA256_RSA_X9_31	YES	LEGACY less than 2048 bit
CKM_SHA384_RSA_X9_31	YES	LEGACY less than 2048 bit
CKM_SHA512_RSA_X9_31	YES	LEGACY less than 2048 bit
CKM_RSA_PKCS_PSS	YES	LEGACY less than 2048 bit
CKM_SHA1_RSA_PKCS_PSS	YES	LEGACY
CKM_SHA224_RSA_PKCS	YES	LEGACY less than 2048 bit
CKM_SHA224_RSA_PKCS_PSS	YES	LEGACY less than 2048 bit
CKM_SHA256_RSA_PKCS	YES	LEGACY less than 2048 bit
CKM_SHA256_RSA_PKCS_PSS	YES	LEGACY less than 2048 bit
CKM_SHA384_RSA_PKCS	YES	LEGACY less than 2048 bit
CKM_SHA384_RSA_PKCS_PSS	YES	LEGACY less than 2048 bit
CKM_SHA512_RSA_PKCS	YES	LEGACY less than 2048 bit
CKM_SHA512_RSA_PKCS_PSS	YES	LEGACY less than 2048 bit

DSA FIPS Mechanisms

DSA FIPS Mechanism	FIPS	Changes in FIPS mode
CKM_DSA_KEY_PAIR_GEN	YES	LEGACY
CKM_DSA	YES	LEGACY
CKM_DSA_PARAMETER_GEN	YES	LEGACY
CKM_SHA1_DSA	YES	LEGACY
CKM_SHA224_DSA	YES	LEGACY
CKM_SHA256_DSA	YES	LEGACY

ECDSA Mechanisms

ECDSA Mechanism	FIPS	Changes in FIPS mode
CKM_EC_KEY_PAIR_GEN	YES	LEGACY for n < 224
CKM_ECDSA	YES	LEGACY for n < 224
CKM_SHA1_ECDSA	YES	LEGACY
CKM_SHA224_ECDSA	YES	LEGACY for n < 224
CKM_SHA256_ECDSA	YES	LEGACY for n < 224
CKM_SHA384_ECDSA	YES	LEGACY for n < 224
CKM_SHA512_ECDSA	YES	LEGACY for n < 224

HMAC Mechanisms

HMAC Mechanism	FIPS	Changes in FIPS mode
CKM_HMAC_SHA224	YES	LEGACY for key length less than 112 bits
CKM_HMAC_SHA256	YES	LEGACY for key length less than 112 bits
CKM_HMAC_SHA384	YES	LEGACY for key length less than 112 bits
CKM_HMAC_SHA512	YES	LEGACY for key length less than 112 bits
CKM_HMAC_SHA1	YES	LEGACY for key length less than 112 bits – ALSO HMAC based KDF is acceptable using an approved hash function including SHA-1

Diffie-Hellman Mechanisms

Diffie-Hellman Mechanisms	FIPS	Changes in FIPS mode
CKM_ECDH1_DERIVE	YES	LEGACY, for n < 224
CKM_ECDH1_COFACTOR_DERIVE	YES	LEGACY, for n < 224

Other Effects

In addition to acceptable key sizes, some algorithms now limit the size of data that can be processed. For example, RSA sign/verify operations, even with sufficiently large key sizes selected, will not run if the input data chunk is too small, when FIPS mode is active. If using an application that is unaware of FIPS-mode limitations, you might encounter errors if you do not adjust the instructions. Using multitoken, as an example, allowing it to use its default data size of 16 bytes, you might see something like this:

```
C:\Program Files\SafeNet\LunaClient>multitoken.exe -mode rsasigver -key 2048 -slots 1
Initializing library...Finished Initializing
...done.
Do you wish to continue?
```

```

Enter 'y' or 'n': y
Constructing thread objects.
Logging in to tokens...
slot 2... Enter password:
Serial Number 151363
Please wait, creating test threads.
Error 0x21 (CKR_DATA_LEN_RANGE) on C_Sign
Aborting tests due to error 0x00000021 (CKR_DATA_LEN_RANGE) on thread 0, slot 1, serial number
150022!
Waiting for threads to terminate.

```

You would correct by including the additional parameter "-packet 32" in the command.

```

C:\Program Files\SafeNet\LunaClient>multitoken -mode rsasigver -key 2048 -slots 1 -packet 32
Initializing library...Finished Initializing
...done.

```

Do you wish to continue?

```

Enter 'y' or 'n': y

Constructing thread objects.
Logging in to tokens...
  slot 1... Enter password: *****
    Serial Number 150022

```

Please wait, creating test threads.

Test threads created successfully. Press ENTER to terminate testing.

RSA sign/verify 2048-bit : (packet size = 32 bytes)

	operations/second		elapsed
1, 0	total	average	time (secs)
-----	-----	-----	-----
111.2	111.2	111.259*	45
111.2	111.2	111.253*	50

Waiting for threads to terminate.

```

C:\Program Files\SafeNet\LunaClient>

```

Modification to DES3 Algorithm for NIST Compliance

In accordance with NIST document SP 800-131A Revision 1, when the HSM is in FIPS mode, two-key DES3 is now restricted to legacy operations (Decryption, Unwrapping, and CMAC verification). All other operations for DES3 must use the three-key variant.

If you are still using Two-key Triple DES, we suggest that you begin adapting your operational work-flow for the following changes that are in effect as of year 2015.

- Encryption, Disallowed
- Decryption, Legacy
- Wrapping, Disallowed
- Unwrapping, Legacy

- CMAC Sign, Disallowed
- CMAC Verification, Legacy

Common Criteria

If you are concerned with Common Criteria validation for products that you use, this section has some information that might be useful in your decisions and planning.

Background

Common Criteria is an international standard for computer security certification that is becoming important worldwide (<http://www.commoncriteriaportal.org>). CC includes categories that are applicable to SafeNet HSM products. The process to have a product validated against CC rules is lengthy and expensive, so we have tended to submit our products when they are mature and when there is a demand.

The process is quite different from US government's FIPS validation, but from the perspective of SafeNet products it has this in common: the unit that is validated is the HSM. That is, a SafeNet Network HSM (or other) appliance would not receive FIPS or CC validation - rather the HSM card that is the core of the appliance, and is used in several products, is what actually gets validated, and only for a particular firmware version. In FIPS evaluation, the HSM card at a certain firmware version is validated. In CC evaluation, the HSM card, as it is used in the SafeNet appliance is evaluated, a subtle difference.

Due to market requirements, the K5 generation (with firmware 4.6.1) as used in SafeNet Network HSM 4.x was CC evaluated and achieved certification. You can check the Common Criteria portal (see link above), or contact SafeNet to find out the most current CC status of any of our products. The K5 is used in the 4.x series 1U rack-mount-format SafeNet Network HSM, and is also the core of some other SafeNet HSM products.

The Common Criteria evaluation process has been started for the **K6** HSM. The earliest expected result would be some time in 2014. From time to time, you can check with the CC site, or with your SafeNet representative to learn the status of SafeNet products in various evaluations.

Trade-offs

As a general practice, when a product receives one of the major validations, and our customers invest their resources in that version of the product, we try to keep supporting that version even while the "state of the art" advances. Thus, as the SafeNet Network HSM product advances (newer software and firmware versions being released) it would be our general practice to make available (sell) the "frozen" CC'd version as long as possible for customers who wish to match units that they already own, while also offering the newer versions for those customers who did not require CC compliance.

What you might notice is that - when possible - a newer release of software is made compatible with an older milestone release of firmware (such as the Common Criteria-validated or FIPS-validated version). Features or fixes to the overall system can take the form of software-only, firmware-only, and combinations that involve changes to both software and firmware to achieve the fix or the new functionality. Thus, it becomes apparent that, if you need to retain strict Common Criteria compliance, you cannot take advantage of any fix or any functional improvement (feature) that has a firmware component to it [since it is the specific hardware and firmware combination that is evaluated and certified].

If you choose to use newer software (Client, Appliance, or both) while retaining an older firmware, be aware that only those new features (and fixes) that are implemented entirely in software would become available to you. Any newer product features that depended on newer firmware could not be accessed without updating the firmware, which would then invalidate that HSM from its strict compliance with Common Criteria - it would no longer be the exact version that was validated. One example is when new encryption algorithms are added, they are implemented in firmware. The

software might allow you to call for a new algorithm, but if you have retained older firmware that didn't include such an algorithm, the response (naturally) would be an error message, such as "Mechanism Invalid".



Note: Due to the way that CC rules work, a validated product must be shipped from the factory. If you already own a SafeNet appliance that has the proper hardware, you cannot simply apply an upgrade/update and achieve CC compliance. Naturally, any competing product faces the same constraints.

So, What Are the Options?

If you absolutely need CC-validated equipment, because your own organization's rules require it, or because your customers require it, then you should purchase a CC-validated version.

If you simply prefer the peace-of-mind associated with the CC blessing - unbiased third-party confirmation that we have conformed to certain rigorous standards with our product - but you also require the newer features or algorithms, consider the following possibility. You might accept that there is some "halo" effect attached to our other products, especially in the same product lines that have been submitted and validated, because we follow the same procedures in our design, testing, sourcing, manufacturing, and other handling for all our HSM products that we do for those that go into the Common Criteria submission pipeline.

In other words, we believe that any given product that we make is likely to meet a CC standard if submitted because we make them all that way. Then we select the one(s) that we believe are early enough in their life-window and their customer appeal to be worth submitting to the year(s?)-long evaluation process.