Luna HSMs and Java PKCS#11 Providers

Integration Guide

Preface

Part Number: 007-011101-001(Rev B, 04/2010)

SafeNet, Inc.

4690 Millennium Drive
Belcamp, Maryland 21017
USA

Limitations

This document does not include the steps to set up the third-party software. The steps given in this document must be modified accordingly. Refer to Luna HSMs documentation for general Luna setup procedures.

Disclaimers

The foregoing integration was performed and tested only with the specific versions of equipment and software and only in the configuration indicated. If your setup matches exactly, you should expect no trouble, and Customer Support can assist with any missteps. If your setup differs, then the foregoing is merely a template and you will need to adjust the instructions to fit your situation. Customer Support will attempt to assist, but cannot guarantee success in setups that we have not tested.

Technical Support

If you encounter a problem while installing, registering or operating this product, please make sure that you have read the documentation. If you cannot resolve the issue, please contact your supplier or SafeNet support.

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

Technical Support Contact Information:
Phone: 800-545-6608, 410-931-7520
Email: support@safenet-inc.com

THIS PAGE INTENTIONALLY LEFT BLANK

Table of Contents

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 1
Introduction

This document covers the necessary information to install, configure and integrate various Java PKCS#11 providers with SafeNet Luna family of general purpose Hardware Security Modules (HSMs) into Java™ applications.

IAIK PKCS#11 Provider

IAIK PKCS#11 Provider provides a simple java.security.keyStore view of these tokens and makes cryptographic operations of these devices accessible via the JCA/JCE framework. For the application, it is just like working with pure software crypto and file key stores. The library accesses the hardware tokens via the PKCS#11 API, also known as Cryptoki. Hence, it can work with any product which supports PKCS#11.

The IAIK JCE Provider for PKCS#11 provides cryptographic functionality, including hash functions, message authentication codes, symmetric, asymmetric, stream encryption, block encryption, key and certificate management. It makes most of the functionality of the PKCS#11 standard accessible to Java™ applications through the JCE API from SUN.

Sun PKCS#11 Provider

The Sun PKCS#11 provider, does not implement cryptographic algorithms itself. Instead, it acts as a bridge between the Java JCA and JCE APIs and the native PKCS#11 cryptographic API, translating the calls and conventions between the two. This means that Java applications calling standard JCA and JCE APIs can, without modification, take advantage of algorithms offered by the underlying PKCS#11 implementations. The Sun PKCS#11 provider supports a number of algorithms, provided that the underlying PKCS#11 implementation offers them.

This enables developers to use cryptographic hardware within their Java™ applications. Applications which are already based on a pure software implementation of the JCE API can use cryptographic hardware with little or no change of their existing applications.

All cryptographic tokens with PKCS#11 version 2.x compliant drivers can be used; this includes cryptographic smart cards and USB tokens as well as hardware security modules (HSMs). In particular, we have tested with SafeNet Luna SA v4.4.1 / Luna PCI v3.0 / Luna CA v2.4.

Scope

3rd Party Application Details

- SUN Java™ Application

Supported Platforms

The following platforms are supported for Luna SA 1U v4.4.1, Luna PCI v3.0, Luna CAv2.4

- RHEL 5.3 X86-64
- Solaris 10 SPARC
- Windows Server 2003 SP2 Enterprise Edition (32-bit)

HSMs and Firmware Version

- K5 HSM f/w 4.6.8(LunaSA)
- K5 HSM f/w 4.7.1(Luna PCI)

- G4 Base f/w 4.6.8(Luna CA4)

Library and Driver Support

- PKCS#11 v2.01 dynamic library
- IAIK PKCS#11 Provider
- Sun PKCS#11 Provider

Distributions

- Luna SA 1U Appliance s/w v4.4.1
- Luna SA Client s/w v4.4.1
- Luna PCI Client s/w v3.0
- Luna PCM Client s/w v2.4

Prerequisites:

Luna SA Setup:

Please refer to the **Luna SA** documentation for installation steps and details regarding to configure and setup the box on UNIX and Windows systems. Before you get started ensure the following:

- Luna SA appliance a secure admin password
- Luna SA a hostname, suitable for your network
- Luna SA network parameters are set to work with your network
- Initialized the HSM on the Luna SA appliance.
- Created and exchanged certificates between the Luna SA and your "Client" system.
- Created a partition on the HSM that will be later used by the SafeNet sample programs. Register the Client with the partition. And run the "vtl verify" command on the client system to display a partition from Luna SA. The general form of command is “/usr/lunasa/bin/vtl verify”.
- Enabled Partition "Activation" and "Auto Activation" (Partition policy settings 22 and 23 (applies to Luna SA with Trusted Path Authentication [which is FIPS 140-2 level 3] only).

Luna PCI Setup:

Please refer to the **Luna PCI** documentation for installation steps and details regarding configuring and setting up the box on UNIX and Windows systems. Before you get started ensure the following:

- Initialize the HSM on the Luna PCI appliance
- Create a partition on the HSM that will be later used by the SafeNet sample programs..
- Enable Partition "Activation" and "Auto Activation" (Partition policy settings 22 and 23 (applies to Luna PCI with Trusted Path Authentication [which is FIPS 140-2 level 3] only).

Luna PCM Setup:

Please refer to the **Luna PCM** documentation for installation steps and details regarding configuring and setting up the box on UNIX and Windows systems. Before you get started ensure the following:

- Initialize the HSM on the Luna CA4 token.
- Create a partition on the HSM that will be later used by the SafeNet sample programs..
- Enable Partition "Activation" and "Auto Activation" (Partition policy settings 22 and 23 (applies to Luna PCI with Trusted Path Authentication [which is FIPS 140-2 level 3] only).

Chapter 2
PKCS#11 Provider Implementation with Luna HSMs

IAIK PKCS#11 Provider

To run the demo programs provided by IAIK PKCS#11 provider with SafeNet Luna SA / Luna PCI / Luna CA4 library:

Install the IAIK PKCS # 11 Provider

1. Include the jar files iaikPkcs11Provider.jar, iaikPkcs11Wrapper.jar and iaik_jce.jar in your class path or put them in the jre/lib/ext directory of you Java runtime (use the signed versions [lib-signed directory] of iaikPkcs11Provider.jar and iaik_jce.jar for JDK 1.4 and higher or if you use JCE 1.2.1 [e.g. IBM JDK 1.3]).

2. Put the shared library libpkcs11wrapper.so for UNIX, in any directory of your system's library search path or VM library path. You can set the VM library path using the java.library.path system property (e.g. using the VM command line argument -Djava.library.path=/home/IAIK_Tools/IAIK_conversion/)

   Alternatively, you can specify the absolute path to the library in your properties file with the key PKCS11_WRAPPER_PATH, eg. PKCS11_WRAPPER_PATH = /home/IAIK_Tools/IAIK_conversion/ libpkcs11wrapper.so.

Configure the Provider with SafeNet Luna SA / Luna PCI HSM

1. Create a properties file called iaik/pkcs/pkcs11/provider/IAIKPkcs11.properties which contains the configuration for your provider (e.g. SafeNet Luna SA HSM).This properties file must contain at least one property entry with the key PKCS11_NATIVE_MODULE.

   Its value must be the PKCS#11 module of the crypto hardware; e.g. PKCS11_NATIVE_MODULE = libCryptoki2_64.so

   You may need to specify the PKCS#11 module with full path name, if it is not in your system's search path. The name of the PKCS#11 module may vary depending on your crypto hardware. Other entries are optional.

2. Put the configuration file somewhere in the class path taking the subdirectory structure into account (same as for class files). You may also include this configuration file in any jar file which is in your class path.

Test the Provider Configuration

Try to run an application that makes a simple test with the provider. Here we have tested SafeNet Luna SA/PCI/CA4 HSM with the demo programs provided by IAIK PKCS#11 provider.

http://jce.iaik.tugraz.at/sic/Products/Core-Crypto-Toolkits/PKCS-11-Provider/demos

You may try some of the included demos; e.g. demo.pkcs.pkcs11.provider.RSASigningDemo0 in the demo directory.

The demo directory contains several samples that show how to use this library. The demos read their PKCS#11 module configuration from properties files. These properties files reside under the resources directory of the respective demo directory. If you do not launch the demos with the included shell files in the test subdirectory, you must ensure that the resources directory is included in the CLASSPATH. Otherwise, the demo will throw an exception saying something like iaik.pkcs.pkcs11.provider.IAIKPkcs11Exception: Required PKCS11_NATIVE_MODULE property has not been configured; e.g. check if properties files are in the CLASSPATH.

In addition, you should not forget to edit the resources/iaik/pkcs/pkcs11/provider/IAIKPkcs11.properties file and enable the appropriate PKCS11_NATIVE_MODULE entry for your PKCS#11 module, in our case it is libCryptoki2_64.so.

These are some crypto operations that we have tested using IAIK PKCS #11 Provider with SafeNet Luna SAv4.4 (PKI Bundle), Luna PCI v3.0 (KE Mode), Luna CA4 v 2.4.

SafeNet provides some demo programs for testing of the following functionality with IAIK pkcs#11 provider.

i. Symmetric Key Generation

ii. RSA Key Pair Generation

iii. Wrapping of RSA Private Key from the symmetric key

iv. Luna CA4 does not support wrapping private keys off the token.

iv. Decrypting/unwrapping the wrapped key from the symmetric Key

v. Sign/Verify using RSA key pair

vi. Encryption and Decryption using Secret Key

vii. Multi-slot support for multiple partitions on a Luna SA or multiple Luna PCM or PCI tokens.

viii. Client authorization for SSL

The Safenet demo programs are available at Customer connection center (http://c3.safenet-inc.com). Look for "*Sample programs for IAIK provider*".

To test Client authorization for SSL use demo programs provided by IAIK.

http://jce.iaik.tugraz.at/sic/Products/Core-Crypto-Toolkits/PKCS-11-Provider/demos

Run SafeNet demo programs:

Use Readme.txt file for safenet demo details.

1. These demo programs are tested on java version "1.6.0_18-ea" with RHEL 5.3 X86-64 and Solaris 10 SPARC with SafeNet Luna SA 4.4.1 with PKI bundle, Luna PCI v3.0 (KE Mode), Luna CA4 v2.4.

2. Include the jar files iaik_javax_crypto.jar, iaik_jce.jar, iaik_jce_full.jar, iaikPkcs11Provider.jar iaikPkcs11Wrapper.jar in your class path.

3. Put the shared library libpkcs11wrapper.so for UNIX, in any directory of your system's library search path or VM library path.

4. Please note that these demos use dynamic provider registration. So don't need to update IAIKPkcs11.properties file.

5. Run the SafeNet demos

   E.g.: Encryption and decryption using Triple – DES algorithm on Luna SA HSM

   # javac –classpath ./iaik_javax_crypto.jar:./iaik_jce.jar:./iaik_jce_full.jar:./iaikPkcs11Provider.jar:./iaikPkcs11Wrapper.jar:./IAIKPkcs11.properties: enc_dec_3des.java

   # java -Djava.library.path=/home/IAIK_Tools/IAIK_conversion/ -cp ./iaik_javax_crypto.jar:./iaik_jce.jar:./iaik_jce_full.jar:./iaikPkcs11Provider.jar:./iaikPkcs11Wrapper.jar:./IAIKPkcs11.properties: enc_dec_3des

Sun PKCS#11 Provider

To run the sample programs provided by Sun PKCS#11 provider with SafeNet Luna SA / Luna PCI / Luna CA4 library:

Install and Configure the Sun PKCS#11 Provider

1. The Sun PKCS#11 provider is implemented by the main class sun.security.pkcs11.SunPKCS11 and accepts the full pathname of a configuration file as an argument. To use the provider, you must first install it by using the Java Cryptography Architecture (JCA).

2. As with all JCA providers, installation of the provider can be done either statically or programmatically. We have tested the sample programs by installing the provider statically. To install the provider statically, add the provider to the Java Security properties file ($JAVA_HOME/lib/security/java.security). For example, here's a fragment of the java.security file that installs the Sun PKCS#11 provider with the configuration file ($JAVA_HOME/lib/security/luna.cfg.

   security.provider.1=sun.security.provider.Sun
   security.provider.2=sun.security.rsa.SunRsaSign
   security.provider.3=com.sun.net.ssl.internal.ssl.Provider
   security.provider.4=com.sun.crypto.provider.SunJCE
   security.provider.5=sun.security.jgss.SunProvider
   security.provider.6=com.sun.security.sasl.Provider
   security.provider.7=sun.security.pkcs11.SunPKCS11 C:\\Program Files\\Java\\jdk1.6.0_18\\jre\\lib\\security\\luna.cfg
   security.provider.8=org.jcp.xml.dsig.internal.dom.XMLDSigRI
   security.provider.9=sun.security.smartcardio.SunPCSC
   security.provider.10=sun.security.mscapi.SunMSCAPI

3. To use more than one slot per PKCS#11 implementation, or to use more than one PKCS#11 implementation, simply repeat the installation for each with the appropriate configuration file. This will result in a Sun PKCS#11 provider instance for each slot of each PKCS#11 implementation.

   security.provider.1=sun.security.provider.Sun
   security.provider.2=sun.security.rsa.SunRsaSign
   security.provider.3=com.sun.net.ssl.internal.ssl.Provider
   security.provider.4=com.sun.crypto.provider.SunJCE
   security.provider.5=sun.security.jgss.SunProvider
   security.provider.6=com.sun.security.sasl.Provider
   security.provider.7=sun.security.pkcs11.SunPKCS11 C:\\Program Files\\Java\\jdk1.6.0_18\\jre\\lib\\security\\luna1.cfg
   security.provider.8=sun.security.pkcs11.SunPKCS11 C:\\Program Files\\Java\\jdk1.6.0_18\\jre\\lib\\security\\luna2.cfg
   security.provider.9=org.jcp.xml.dsig.internal.dom.XMLDSigRI
   security.provider.10=sun.security.smartcardio.SunPCSC
   security.provider.11=sun.security.mscapi.SunMSCAPI

4. Create a configuration file which contains the configuration for your provider. The configuration file is a text file that contains entries in the following format.

   attribute = value

The valid values for attribute and value are described in the table in this section. The two mandatory attributes are name and library. Here is a sample configuration file. The values provided in the sample configuration file can vary depending upon the configuration.

```
#SafeNet Luna

name = Luna
library = C:\Program Files\LunaSA\cryptoki.dll
description = Luna config
slot = 1

attributes(*,*,*) = {
CKA_TOKEN = true
}

attributes(*,CKO_SECRET_KEY,*) = {
CKA_CLASS=4
CKA_PRIVATE= true
CKA_KEY_TYPE = 21
CKA_SENSITIVE= true
CKA_ENCRYPT= true
CKA_DECRYPT= true
CKA_WRAP= true
CKA_UNWRAP= true
}

attributes(*,CKO_PRIVATE_KEY,*) = {
CKA_CLASS=3
CKA_LABEL=true
CKA_PRIVATE = true
CKA_DECRYPT=true
CKA_SIGN=true
CKA_UNWRAP=true
}

attributes(*,CKO_PUBLIC_KEY,*) = {
CKA_CLASS=2
CKA_LABEL=true
CKA_ENCRYPT = true
CKA_VERIFY=true
CKA_WRAP=true
}
```

Test the Provider Configuration

We have tested the following APIs mentioned below with SafeNet Luna SA/PCI/CA4 HSM using some sample programs utilizing the Sun PKCS#11 provider.

i. Symmetric Key Generation

ii. RSA Key Pair Generation

iii. Sign/Verify using RSA key pair

iv. Encryption and Decryption using Secret Key

v. Multi-slot support for multiple partitions on a Luna SA or multiple Luna PCM or PCI tokens.

vi. Client authorization for SSL

The sample programs are available at Customer connection center (http://c3.safenet-inc.com). Look for "*Sample programs for Sun PKCS#11 provider*".

Run SafeNet demo programs:

1. To test MultiSlot support for multiple tokens:

   #java MultiSlot.java

References

1. Details about demos in IAIK demo directory
   http://jce.iaik.tugraz.at/sic/Products/Core-Crypto-Toolkits/PKCS-11-Provider/demos
2. Download center for IAIK provider jars
   http://jce.iaik.tugraz.at/sic/Download
3. Java™ PKCS#11 Reference Guide
   http://java.sun.com/javase/6/docs/technotes/guides/security/p11guide.html
4. SafeNet general purpose HSMs, Network attached
   http://www.safenet-inc.com/Products/Data_Protection/Hardware_Security_Modules/General_Purpose_HSMs,_Network_Attached.aspx