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

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

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

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

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

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

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

Gemalto hereby disclaims all warranties and conditions with regard to the information contained herein, including all implied warranties of merchantability, fitness for a particular purpose, title and non-infringement. In no event shall Gemalto be liable, whether in contract, tort or otherwise, for any indirect, special or consequential damages or any damages whatsoever including but not limited to damages resulting from loss of use, data, profits, revenues, or customers, arising out of or in connection with the use or performance of information contained in this document.

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

© 2012-18 Gemalto. All rights reserved. Gemalto and the Gemalto logo are trademarks and service marks of Gemalto and/or its subsidiaries and are registered in certain countries. All other trademarks and service marks, whether registered or not in specific countries, are the property of their respective owners.

Document Part Number: 007-009988-001, Rev. N
Release Date: August 2018
Preface .................................................................................................................................. 4
Scope ........................................................................................................................................... 4
Document Conventions .................................................................................................................. 4
Command Syntax and Typeface Conventions ................................................................................ 5
Support Contacts .......................................................................................................................... 6

1 Introduction .............................................................................................................................. 7
Overview ........................................................................................................................................ 7
3rd Party Application Details ......................................................................................................... 8
Supported Platforms ..................................................................................................................... 8
Prerequisites ................................................................................................................................ 8
Configuring SafeNet Luna Network HSM ..................................................................................... 8
Using SafeNet HSM in FIPS Mode ............................................................................................... 9
Provision your HSM on Demand Service ...................................................................................... 9
Setup Microsoft Authenticode ....................................................................................................... 10

2 Integrating SafeNet HSM with Microsoft Authenticode ............................................................. 11
Configuring the SafeNet Cryptographic Storage Provider ........................................................... 11
Signing and Time Stamping the code ........................................................................................... 12

3 Integrating SafeNet HSM with MS Strong Name ...................................................................... 13
Configuring the SafeNet Cryptographic Storage Provider ........................................................... 13
Signing a .NET Assembly ............................................................................................................. 15

4 Integrating SafeNet HSM with MS Mage/ClickOnce ................................................................ 17
Configuring the SafeNet Cryptographic Storage Provider ........................................................... 17
Deploying an application with the Mage.exe command line tool ................................................... 18

5 Integrating SafeNet HSM with Microsoft Hardware Certification Kit (HCK) ............................ 21
Configuring the SafeNet Cryptographic Storage Provider ........................................................... 21
Signing the package using the CA signed certificate .................................................................... 22
Singing the package using a self-signed certificate ....................................................................... 23

6 Troubleshooting Tips ................................................................................................................ 30
Troubleshooting .......................................................................................................................... 30
Problem ....................................................................................................................................... 30
Solution ....................................................................................................................................... 30
Preface

This document is intended to guide security administrators to install, configure, and integrate Microsoft Authenticode with SafeNet Luna Hardware Security Module (HSM) or an HSM on Demand service.

Scope

This document covers the necessary information to install, configure, and integrate Microsoft Authenticode with SafeNet Network HSM or HSM on Demand service.

Document Conventions

This section provides information on the conventions used in this template.

Notes

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

CAUTION: Exercise caution. Caution alerts contain 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. These elements 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

<table>
<thead>
<tr>
<th>Convention</th>
<th>Description</th>
</tr>
</thead>
</table>
| **bold**   | The bold attribute is used to indicate the following:  
  • Command-line commands and options (Type `dir /p`.)  
  • Button names (Click **Save As**.)  
  • Check box and radio button names (Select the **Print Duplex** check box.)  
  • Window titles (On the **Protect Document** window, click **Yes**.)  
  • Field names (**User Name**: Enter the name of the user.)  
  • Menu names (On the **File** menu, click **Save**.) (Click **Menu > Go To > Folders**.)  
  • User input (In the **Date** box, type **April 1**.) |
| **italic** | The italic attribute is used for emphasis or to indicate a related document. (See the **Installation Guide** for more information.) |
| **Consolas** | Denotes syntax, prompts and code examples. |
## Support Contacts

<table>
<thead>
<tr>
<th>Contact Method</th>
<th>Contact Information</th>
</tr>
</thead>
</table>
| **Address**          | Gemalto  
4690 Millennium Drive  
Belcamp, Maryland 21017, USA                                                       |
| **Phone**            | US 1-800-545-6608  
International 1-410-931-7520                                                   |
| **Technical Support**| **Customer Portal**  
https://supportportal.gemalto.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. |
Overview

This document covers the necessary information to install, configure, and integrate Microsoft Authenticode with SafeNet Luna Hardware Security Modules (HSM) or an HSM on Demand (HSMoD) Service. SafeNet HSMs come as on premise hardware HSMs widely known as SafeNet Luna HSM and a Data Protection on Demand (DPoD) cloud offering HSM on Demand Service.

Microsoft Authenticode permits end users to identify who has published a software component and verify that no one has tampered with it before downloading it from the Internet. Authenticode guarantees the integrity of published software.

Authenticode relies on proven cryptographic techniques and the use of one or more private keys to sign and time-stamp the published software. It is important to maintain the confidentiality of these keys. SafeNet Hardware Security Module (HSM) or an HSMoD service integrates with Microsoft Authenticode to provide a trusted system for protecting the organizational credentials of the software publisher. SafeNet HSMs secures the code-signing key within an industry standard FIPS 140-2 level 3 validated HSM.

The benefits of using SafeNet HSMs with Authenticode include:

- Secure generation, storage and protection of the Identity signing private key on FIPS 140-2 level 3 validated hardware*.
- Full life cycle management of the keys.
- HSM audit trail.

**NOTE:** HSM on Demand services do not have access to the secure audit trail.

- Protection for the organizational credentials of the software publisher.
- Secure storage of the private key.
- Provision of a trusted time-stamp to Authenticode.

*FIPS 140-2 validation in progress for HSMoD services

This integration guide contains the following topics:

- Integrating SafeNet HSM with Microsoft Authenticode
- Integrating SafeNet HSM with MS Strong Name
- Integrating SafeNet HSM with MS Mage/ClickOnce
- Integrating SafeNet HSM with Microsoft Hardware Certification Kit (HCK)
3rd Party Application Details

- Microsoft Authenticode

Supported Platforms

Below is the list of the platforms tested with the following HSMs:

**SafeNet Luna HSM:** SafeNet Luna HSM appliances are purposefully designed to provide a balance of security, high performance, and usability that makes them an ideal choice for enterprise, financial, and government organizations. SafeNet Luna HSMs physically and logically secure cryptographic keys and accelerate cryptographic processing. The SafeNet Luna HSM on premise offerings include the SafeNet Luna Network HSM, SafeNet PCIe HSM, and SafeNet Luna USB HSMs. SafeNet Luna HSMs are also available for access as an offering from cloud service providers such as IBM cloud HSM and AWS cloud HSM classic. This integration is supported with SafeNet Luna HSM on the following operating systems:

- Windows 2016 Server
- Windows Server 2012R2

**SafeNet Data Protection on Demand (DPOD):** is a cloud-based platform that provides on-demand HSM and Key Management services through a simple graphical user interface. With DPOD, security is simple, cost effective and easy to manage because there is no hardware to buy, deploy and maintain. As an Application Owner, you click and deploy services, generate usage reports and maintain just the services you need. This integration is supported/verified with SafeNet DPOD on the following operating systems:

- Windows 2016 Server
- Windows Server 2012R2

**Prerequisites**

Before starting the integration of Microsoft Authenticode with SafeNet Luna HSM or HSM on Demand Service, ensure you have completed configuring the SafeNet Luna Network HSM or provisioning HSM on Demand Service as per the requirement.

**Configuring SafeNet Luna Network HSM**

Before you get started ensure the following:

1. SafeNet Luna Network HSM appliance and a secure admin password.
2. SafeNet Luna Network HSM, and a hostname, suitable for your network.
3. SafeNet Luna Network HSM network parameters are set to work with your network.
4. Initialize the HSM on the SafeNet Luna Network HSM appliance.
5. Create and exchange certificates between the SafeNet Luna Network HSM and your Client system.
6. Create a partition on the HSM that will be later used by Microsoft Authenticode.

NOTE: Microsoft Authenticode Integration is tested with Luna Clients in FIPS and HA mode also.
7. Register a client for the system and assign the client to the partition to create an NTLS connection. Initialize Crypto Officer and Crypto User roles for the registered partition.

8. Ensure that the partition is successfully registered and configured. The command to see the registered partition is:

```bash
# lunacm.exe
```

Example output:

```
Available HSMs:

  Slot Id -> 0
  Label -> HSM7
  Serial Number -> 1213475834614
  Model -> LunaSA 7.3.0
  Firmware Version -> 7.3.0
  Configuration -> Luna User Partition With SO (PW) Signing With Cloning Mode
  Slot Description -> Net Token Slot
  Current Slot Id: 0
```

**NOTE:** Follow the SafeNet Network Luna HSM documentation for detailed steps for creating NTLS connection, initializing the partitions and various user roles.

---

**Using SafeNet HSM in FIPS Mode**

Under FIPS 186-3/4, the RSA methods permitted for generating keys are 186-3 with primes and 186-3 with aux primes. This means that RSA PKCS and X9.31 key generation is no longer approved for operation in a FIPS-compliant HSM. If you are using the SafeNet HSM in FIPS mode, you have to make the following change in configuration file:

```
[Misc]
RSAKeyGenMechRemap = 1
```

The above setting redirects the older calling mechanism to a new approved mechanism when the SafeNet HSM is in FIPS mode.

**Provision your HSM on Demand Service**

This service provides your client machine with access to an HSM Application Partition for storing cryptographic objects used by your applications. Application partitions can be assigned to a single client, or multiple clients can be assigned to, and share, a single application partition.

To use the HSM on Demand service you need to provision your application partition, starting by initializing the following roles:

- **Security Officer (SO)** - responsible for setting the partition policies and for creating the Crypto Officer.
- **Crypto Officer (CO)** - responsible for creating, modifying and deleting crypto objects within the partition. The CO can use the crypto objects and create an optional, limited-capability role called Crypto User that can use the crypto objects but cannot modify them.
- **Crypto User (CU)** - optional role that can use crypto objects while performing cryptographic operations.

**NOTE:** Refer to the “SafeNet Data Protection on Demand Application Owner Quick Start Guide” for procedural information on configuring the HSM on Demand service and create a service client.

The HSM on Demand service client package is a zip file that contains system information needed to connect your client machine to an existing HSM on Demand service.

**Constraints on HSMoD Services**

**HSM on Demand Service in FIPS mode**

HSMoD services operate in a FIPS and non-FIPS mode. If your organization requires non-FIPS algorithms for your operations, ensure you enable the **Allow non-FIPS approved algorithms** check box when configuring your HSM on Demand service. The FIPS mode is enabled by default.

Refer to the “Mechanism List” in the SDK Reference Guide for more information about available FIPS and non-FIPS algorithms.

**Verify HSM on Demand <slot> value**

LunaCM commands work on the current slot. If there is only one slot, then it is always the current slot. If you are completing an integration using HSMoD services, you need to verify which slot on the HSMoD service you send the commands to. If there is more than one slot, then use the **slot set** command to direct a command to a specified slot. You can use slot list to determine which slot numbers are in use by which HSMoD service.

**Setup Microsoft Authenticode**

**Installing the Windows SDK**

The Authenticode programs (makecert, cert2spc, etc.,) are installed with Microsoft Visual Studio and Microsoft Windows SDK.

Refer to the Microsoft Windows SDK installation documentation for further information.

**Installing the Office Smart Tags SDK**

To demonstrate the Authenticode technology, this integration guide requires the Microsoft Office Smart Tags SDK.

Refer to the Microsoft Office Smart Tags SDK installation documentation for further information.
Integrating SafeNet HSM with Microsoft Authenticode

Microsoft Authenticode permits end users to identify who published a software component and verify that no one has tampered with the software component before downloading the object from the internet.

This integration guide contains the following topics:

• Configuring the SafeNet Cryptographic Storage Provider
• Signing and Time Stamping the code

Configuring the SafeNet Cryptographic Storage Provider

To use Microsoft Authenticode with a SafeNet HSM configure the SafeNet Cryptographic Service Provider (CSP) to generate the certificates for Microsoft Authenticode.

To configure the SafeNet CSP

   a. Run the command, `register.exe` to register Luna CSP. The general form of command is given below:
      `<Luna Client Installation Directory>\CSP>register.exe`
   b. Register the Luna Cryptographic Services for Microsoft Windows. The general form of command is given below:
      `<Luna Client Installation Directory>\CSP>register.exe /1`

2. Generate a certificate using the `makecert` command and the Luna CSP "Luna Cryptographic Services for Microsoft Windows".
   `makecert -sk mykey -sp "Luna Cryptographic Services for Microsoft Windows" -n "CN=Common Name" -r -ss mystore Test.cer`
   where:
   - `sk` The location of the subject's key container which holds the private key.
   - `sp` Subject CryptoAPI's provider name.
   - `n` The name and details of the publisher's certificate.
   - `ss` The name of the subject's certificate store in which the generated certificate will be stored.

3. Create a Software Publishing Certificate (SPC) from the generated certificate.
   `cert2spc Test.cer Test.spc`
Signing and Time Stamping the code

Sign and time stamp the code using the signtool sign command and the Software Publishing Certificate (SPC).

To sign and time stamp the code

1. Sign and time stamp the code using signtool:

   ```
signtool sign /v /f Certificate /p Pin /csp "Cryptographic Service Provider Name" /k "Key Container Name" /t timestamp URL "File to be signed"
   
   where:
   
   /f Publisher’s Certificate.
   /p HSM partition password.
   /k Container Name that contains the signing key.
   /t URL used for Time Stamping.
   ```
Strong Name is the part of Microsoft SDK that offers a powerful mechanism for giving .NET Framework assemblies unique identities. To get a valid strong name, an assembly is strong-name signed during the build process. This is done using the private key that corresponds to the public key in the strong name. The strong name signature can then be verified using the public key.

This integration guide contains the following topics:

- Configuring the SafeNet Cryptographic Storage Provider
- Signing a .NET Assembly

### Configuring the SafeNet Cryptographic Storage Provider

To use Microsoft Strong Name with a SafeNet HSM configure the SafeNet Cryptographic Service Provider (CSP) to generate the keys and certificates for Microsoft Authenticode.

#### To configure the SafeNet Cryptographic Storage Provider

1. Configure Luna Cryptographic Service Provider (CSP) on Windows Server.
   a) Open the command prompt and run `register.exe` to register Luna CSP. The general form of command is given below:
      ```
      <Luna Client Installation Directory>CSP>register.exe
      ```
   b) To register the Luna Cryptographic Services for Microsoft Windows. The general form of command is given below:
      ```
      <Luna Client Installation Directory>CSP>register.exe /l
      ```
2. Generate a certificate using the `makecert` command and the Luna CSP "Luna Cryptographic Services for Microsoft Windows".
   ```
   makecert -sk <keyContainer> -sp "Luna Cryptographic Services for Microsoft Windows" -n "CN=Common Name" -ss <certStore> CertificateName.cer
   ```
   where:
   - `sk` The location of the subject's key container which holds the private key
   - `sp` Subject CryptoAPI's provider name
   - `n` The name and details of the signer's certificate
   - `ss` The name of the subject's certificate store in which the generated certificate will be stored.
3. Make the Luna CSP the default CSP to use with Microsoft Strong Name using the following command:
   `sn.exe -c "Luna Cryptographic Services for Microsoft Windows"`

4. Extract the public key from the key-pair generated in step 2 using the following command:
   `sn.exe -pc mykey mykey.snk`
   where "mykey" is the name of key container and "mykey.snk" is name of public key file.
Signing a .NET Assembly

You can use MS Strong by signing a .NET Assembly.

To sign a .NET assembly

1. Write any C# program. Open the Visual Studio command prompt and compile the program. Delay sign the generated exe file. Use the following command:

   `csc /delaysign+ /keyfile:"C:\Program Files (x86)\Microsoft SDKs\Windows\v8.1A\bin\NETFX 4.5.1 Tools\x64\mykey.snk" C:\Users\Administrator\Desktop\myapp.cs`

   Where "/keyfile" is the public key extracting from the key-pair in the previous command.

2. Sign the generated exe with Strong Name:

   `sn.exe -Rc C:\Users\Administrator\Desktop\myapp.exe mykey`

   Where "mykey" is the key container in which you have generated the key-pair.
3. Verify the assembly is Strong Name signed using the following command:

```
sn.exe -v C:\Users\Administrator\Desktop\myapp.exe
```

![Image of command output]

Strong Name is successfully signed and verified with .Net assembly using the key-pair generated on SafeNet HSM.
Microsoft's Mage.exe is a Manifest Generation and Editing command line Tool for .NET Framework applications. There is also a UI version MageUI.exe. A typical use is manually creating your ClickOnce deployment manifests. This guide assumes that you have a Windows application that you are ready to deploy. This application will be referred to as AppToDeploy.

For more details about MS Mage/ClickOnce signing, refer Microsoft Documentation.

NOTE: SafeNet HSM is used to secure the signing keys so that your signing keys never access by any unauthorized entity. Mage.exe is a 32 bit application so you have to use the 32 bit Luna Client with 32 bit CSP.

This integration guide contains the following topics:

- Configuring the SafeNet Cryptographic Storage Provider
- Deploying an application with the Mage.exe command line tool

### Configuring the SafeNet Cryptographic Storage Provider

To use Microsoft Mage/ClickOnce with a SafeNet HSM configure the SafeNet Cryptographic Service Provider (CSP) to generate the keys and certificates for Microsoft Mage/ClickOnce.

#### To configure the SafeNet CSP

1. Install Luna Cryptographic Service Provider (CSP) on Windows Server.
   a) Open the command prompt and run `register.exe` to register Luna CSP. The general form of command is `<Luna Client Installation Directory>\win32\csp>register.exe`
   b) To register the Luna Cryptographic Services for Microsoft Windows. The general form of command is `<Luna Client Installation Directory>\win32\csp>register.exe /l`

2. Generate a certificate using the `makecert` command and the Luna CSP "Luna Cryptographic Services for Microsoft Windows".
   ```bash
   makecert -sk <keyContainer> -sp "Luna Cryptographic Services for Microsoft Windows" -n "CN=Common Name" -ss <certStore> CertificateName.cer
   ```
   where:
   -`sk` The location of the subject’s key container which holds the private key
   -`sp` Subject CryptoAPI’s provider name
   -`n` The name and details of the signer’s certificate
-ss   The name of the subject’s certificate store in which the generated certificate will be stored. Use "My" which is the default user cert store where the application is looking for certificate.

3. After generating the certificate, use this certificate in Visual Studio to sign the Application/Deployment manifest. Open the Properties window of the project and click on the Signing and then select Sign the ClickOnce manifests. Click on **Select from Store…** and click **OK** after choosing the certificate that was generated in step 1.

---

**Deploying an application with the Mage.exe command line tool**

You can deploy applications using the Mage.exe command line tool.

**To deploy an application with the Mage.exe command line tool**

1. Create a directory to store your **ClickOnce** deployment files.
2. In the deployment directory just created, create a version subdirectory. If this is the first time application is deployed, name the version subdirectory 1.0.0.0.
3. Copy all of application files to the version subdirectory, including executable files, assemblies, resources, and data files. If necessary, create additional subdirectories that contain additional files.
4. Open the Windows SDK or Visual Studio command prompt and change to the version subdirectory.
5. Create the application manifest with a call to Mage.exe. The following statement creates an application manifest for code compiled to run on the msil processor.

```
mage -New Application -Processor msil -ToFile AppToDeploy.exe.manifest -name "MyApp" -Version 1.0.0.0 -FromDirectory .
```

6. Sign the application manifest with Authenticode certificate.

```
mage -Sign AppToDeploy.exe.manifest -CertHash "Certificate Hash"
```

7. Run the certutil.exe to know the certificate hash value the command would be `certutil -verifystore -user My`. 

My is the user cert store where the certificate is generated using the `makecert` command.

```
certutil -verifystore -user My
```

8. Change to the root of the deployment directory.

9. Generate the deployment manifest with a call to Mage.exe. By default, Mage.exe marks your ClickOnce deployment as an installed application, so that it can be run both online and offline. To make the application available only when the user is online, use the `-Install` option with a value of false. If default option is used
and user install the application from a Web site or file share, ensure that the value of the -ProviderUrl option points to the location of the application manifest on the Web server or share.

\mage\ New Deployment -Processor msil -Install true -Publisher "My Company" -ProviderUrl "\\myServer\myShare\AppToDeploy.application" -AppManifest 1.0.0.0\AppToDeploy.exe.manifest -ToFile AppToDeploy.application

10. Sign the deployment manifest with the Authenticode certificate.

\mage\ Sign AppToDeploy.application -CertHash "Certificate Hash"

11. Copy all the files in the deployment directory to the deployment destination or media. This may be either a folder on a Web site or FTP site, a file share, or a CD-ROM.

12. Provide your users with the URL, UNC, or physical media required to install your application. If you provide a URL or a UNC, you must give your users the full path to the deployment manifest. For example, if AppToDeploy is deployed to http://webserver01/ in the AppToDeploy directory, the full URL path would be http://webserver01/AppToDeploy/AppToDeploy.application.
Microsoft’s Windows Certification Program is designed to help your company deliver compatible and reliable systems, software, and hardware products. End users trust the logo as an assurance of compatibility and reliability. This program is intended to help you develop systems and devices that have been tested to ensure that they meet Microsoft standards for Windows 8.1 as well as the quality level that ensures a great Windows experience for end users.

SafeNet HSM is used to secure the signing keys so that your signing keys cannot be accessed by any unauthorized entity. Microsoft HCK uses RSA keys for signing packages.

This integration guide contains the following topics:

- Configuring the SafeNet Cryptographic Storage Provider
- Signing the package using the CA signed certificate
- Singing the package using a self-signed certificate

### Configuring the SafeNet Cryptographic Storage Provider

To use Microsoft Mage/ClickOnce with a SafeNet HSM configure the SafeNet Cryptographic Service Provider (CSP) to generate the keys and certificates for Microsoft Mage/ClickOnce.

**To configure the SafeNet CSP**

1. Install Luna Cryptographic Service Provider (CSP) on Windows Server.
   a. Open the command prompt and run `register.exe` to register Luna CSP. The general form of command is given below:
      
      `<Luna Client Installation Directory>\win32\csp>register.exe`

   b. To register the Luna Cryptographic Services for Microsoft Windows. The general form of command is given below
      
      `<Luna Client Installation Directory>\win32\csp>register.exe /l`

2. To verify the registered cryptographic providers, browse to ”C:\Windows\SysWOW64” and execute ”certutil -csplist”
3. In order to integrate the SafeNet HSM with Microsoft HCK, the Luna CSP Luna Cryptographic Services for Microsoft Windows must be used to generate the certificate. The certificate must be signed and the signer certificate must be present in the "Trusted Root Certificate Authority". You can use the CA signed certificate of self-signed certificate both. There are two methods that you can use to generate the signing certificate:

**Signing the package using the CA signed certificate**

Sign the package using the CA certificate.

**To sign the package using a CA signed certificate**

1. Create an inf file with the following attributes:

   ```
   [Version]
   Signature="$Windows NT$"
   [NewRequest]
   Subject = "C=US,O=SafeNet,CN=HCK,OU=HCKIntegration"
  KeySpec = 1
   KeyLength = 2048
   Exportable = FALSE
   MachineKeySet = TRUE
   KeyContainer = HCK
   ProviderName = "Luna Cryptographic Services for Microsoft Windows"
   ProviderType = 1
   KeyUsage = 0x04
   ```

2. Generate a certificate request using the created inf. Make sure to use the 32 bit `certreq` utility from the "C:\Windows\SysWOW64" directory. A success message is displayed after this command has been executed.

3. Take the generated certificate request to a CA and get it signed to obtain a signed certificate.

4. Now we have to import this obtained certificate in the user's personal certificate store. As this setup is 32 bit, ensure to use the 32 bit Microsoft Certificate manager console.

   ```
   C:\Windows\SysWOW64\certmgr.msc
   ```

5. Right-click on **Personal** -> **All Task** -> **Import** and follow the instruction to import the signed certificate. Verify the certificate is successfully imported.

6. Double-click the certificate and confirm that there is a private key mapped with this certificate. Check the message at the bottom.
7. In case, the private key is not mapped correctly, repair the certificate using the "certutil –repairokstore" utility.
8. Open the certificate.
10. Select the Serial number field.
11. Copy the serial number or thumb print.
12. Execute the "certutil -repairokstore -user My "SerialNumber or ThumbPrint" command from the SysWOW64 directory to map the private key (on the HSM) with the certificate.
13. After the repairokstore command has been successfully executed, refresh the certificate manager snap in, open the certificate and confirm the message at the bottom is displayed.

Singing the package using a self-signed certificate

Sign the package using the CA certificate.

To sign the package using a self-signed certificate

1. Use the makecert utility to generate a self-signed certificate. Browse to the "C:\Program Files (x86)\Windows Kits\8.1\bin\x86" directory and execute the following command:
makecert -sk <keyContainer> -sp "Luna Cryptographic Services for Microsoft Windows" -r -n "CN=Common Name" -ss <certStore> CertificateName.cer

where:
- sk   The location of the subject’s key container which holds the private key
- sp   Subject CryptoAPI’s provider name
- n    The name and details of the signer’s certificate
- ss   The name of the subject’s certificate store in which the generated certificate will be stored. Use "My" which is the default user cert store where the application is looking for certificate.

2. Open the `certmgr.msc` from the "C:\Windows\SysWOW64" directory and export the generated certificate from the Personal folder.

3. Import the certificate in the **Trusted Root Certificate Authority** folder. Verify that certificate imported successfully.
4. Now, when the certificate and the private key is ready for signing, open Windows Hardware Certification Kit and import the project to sign.

5. Navigate through the various tabs to verify the project imported is correct.
5– Integrating SafeNet HSM with Microsoft Hardware Certification Kit (HCK)
6. After verification, go to the package tab and click on create package to sign the package. It prompts for **Signing Options** Select **Use certificate Store** and click **OK**.

7. Select the signing certificate. From the pop up, select the certificate that was imported earlier on the local machine’s personal certificate store and click **OK**.
8. Select a location to save the signed package and click **Save**.

9. Click **Save** to start signing. Signing starts with a **Creating Package** window.
10. In the end, if the certificate and the private key are correctly mapped, a success message displays and you can verify the signed package in the location you saved it.
Troubleshooting

Problem

You encounter the following error when running the makecert command on an HSM in FIPS mode.

**Error:** CryptHashPublicKeyInfo failed => 0x80090005 (-2146893819) Failed.

Solution

The cert always has an MD5 hash in it. Configure the Luna CSP to do MD5 in software. The general form of command is as follows:

```cmd
<Luna Client Installation Directory>CSP > Register.exe /algorithms
```

It prompts you to register the various algorithms; you need to register the MD5 algorithms in software.