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.

© 2009-18 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.
# Contents

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 ........................................................................................................... 7  
Supported Platforms .................................................................................................................. 7  
  SafeNet Luna HSM (v7.x) ........................................................................................................... 7  
  SafeNet Luna HSM (v4.x/5.x/6.x) ............................................................................................... 8  
  SafeNet Luna PCI-E HSM .......................................................................................................... 8  
Prerequisites ....................................................................................................................................... 9  
  Configuring SafeNet Luna Network HSM 7.x ........................................................................... 9  
  Configuring SafeNet Luna Network HSM (v4.x/5.x/6.x) ............................................................ 11  
  SafeNet Luna PCI-E HSM Setup ............................................................................................... 11  
  Using Luna 6.x/7.x in FIPS Mode ............................................................................................ 12  
  Microsoft Authenticode Setup ................................................................................................. 12  

2 Integrating SafeNet Luna HSM with Microsoft Authenticode (Windows SDK for Windows  
  Server 2003 SP1) ........................................................................................................... 13  

3 Integrating SafeNet Luna HSM with Microsoft Authenticode (Windows SDK for Windows  
  Server 2008) .................................................................................................................. 15  

4 Integrating SafeNet Luna HSM with Microsoft Authenticode (Windows SDK for Windows  
  Server 2012 R2/2016) ....................................................................................................... 18  

5 Integrating SafeNet Luna HSM with MS Strong Name (Windows Server 2012 R2) ....... 20  

6 Integrating SafeNet Luna HSM with MS Mage/ClickOnce (Windows Server 2012 R2) .. 24  

7 Integrating SafeNet Luna HSM with Microsoft HCK (Windows Server 2012) ............... 28  
  Method 1 ........................................................................................................................................ 29  
  Method 2 ........................................................................................................................................ 30  

8 Troubleshooting Tips ........................................................................................................... 37  
Troubleshooting ..................................................................................................................... 37  
  Problem ................................................................................................................................. 37  
  Solution ................................................................................................................................. 37
Preface

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

Scope

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

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>
<tbody>
<tr>
<td>Address</td>
<td>Gemalto&lt;br&gt;4690 Millennium Drive&lt;br&gt;Belcamp, Maryland 21017, USA</td>
</tr>
<tr>
<td>Phone</td>
<td>US 1-800-545-6608&lt;br&gt;International 1-410-931-7520</td>
</tr>
<tr>
<td><strong>Technical Support</strong></td>
<td><strong>Customer Portal</strong></td>
</tr>
<tr>
<td><strong><a href="https://supportportal.gemalto.com">https://supportportal.gemalto.com</a></strong></td>
<td>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.</td>
</tr>
</tbody>
</table>
Overview

This document covers the necessary information to install, configure, and integrate Microsoft Authenticode with SafeNet Luna Hardware Security Modules (HSM).

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 assures end users about the identity of the software publisher. It also assures the code has not been altered after the signature was applied, before they download signed code from the Internet.

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 Luna Hardware Security Module (HSM) integrates with Microsoft Authenticode to provide a trusted system for protecting the organizational credentials of the software publisher. SafeNet Luna HSMs secures the code-signing key within an industry standard FIPS 140-2 level 3 validated HSM.

3rd Party Application Details

- Microsoft Authenticode (Microsoft Windows SDK 10.1)
- Microsoft Authenticode (Microsoft Windows SDK 8.1)
- Microsoft Authenticode (Microsoft Windows SDK 6.1)

Supported Platforms

SafeNet Luna HSM (v7.x)

<table>
<thead>
<tr>
<th>Platforms Tested</th>
<th>SafeNet Luna Network HSM</th>
<th>SafeNet Luna HSM Client Version</th>
<th>Microsoft SDK</th>
<th>Microsoft Office Smart Tags SDK (Optional)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows Server 2016</td>
<td>s/w 7.2.0</td>
<td>7.2.0</td>
<td>v10.1</td>
<td>Office 2003 SDK</td>
</tr>
<tr>
<td></td>
<td>f/w 7.2.0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Windows Server 2016</td>
<td>s/w 7.1.0</td>
<td>7.1.0</td>
<td>v10.1</td>
<td>Office 2003 SDK</td>
</tr>
<tr>
<td>Windows Server 2012R2</td>
<td>s/w 7.1.0</td>
<td>7.1.0</td>
<td>v10.1</td>
<td>Office 2003 SDK</td>
</tr>
</tbody>
</table>
## Platforms Tested

<table>
<thead>
<tr>
<th>Platforms Tested</th>
<th>SafeNet Luna Network HSM</th>
<th>SafeNet Luna HSM Client Version</th>
<th>Microsoft SDK</th>
<th>Microsoft Office Smart Tags SDK (Optional)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows Server 2016</td>
<td></td>
<td>7.0.0</td>
<td>v10.1</td>
<td>Office 2003 SDK</td>
</tr>
<tr>
<td>Windows Server 2012R2</td>
<td>s/w 7.0.0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>f/w 7.0.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SafeNet Luna HSM (v4.x/5.x/6.x)</td>
<td></td>
<td>6.3.0</td>
<td>v10.1</td>
<td>Office 2003 SDK</td>
</tr>
<tr>
<td>Windows Server 2016</td>
<td>s/w 6.3.0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Windows Server 2012R2</td>
<td>f/w 6.27.0 /6.10.9</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Windows Server 2012R2</td>
<td>s/w 6.2.2</td>
<td></td>
<td>v8.1</td>
<td>Office 2003 SDK</td>
</tr>
<tr>
<td></td>
<td>f/w 6.24.3 /6.10.9</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Windows Server 2012R2</td>
<td>s/w 6.2.1</td>
<td></td>
<td>v8.1</td>
<td>Office 2003 SDK</td>
</tr>
<tr>
<td></td>
<td>f/w 6.24.2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SafeNet Luna PCI-E HSM</td>
<td></td>
<td>5.2.1</td>
<td>v8.1</td>
<td>Office 2003 SDK</td>
</tr>
<tr>
<td>Windows Server 2012 Standard</td>
<td>s/w 5.2.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Windows Server 2012 R2</td>
<td>f/w 6.10.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Windows Server 2008 (64 and 32 bit)</td>
<td>s/w 5.1</td>
<td></td>
<td>v6.1</td>
<td>Office 2003 SDK</td>
</tr>
<tr>
<td>Windows Server 2008</td>
<td>f/w 6.2.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Windows Server 2003 SP1</td>
<td>s/w 4.4</td>
<td></td>
<td>v6.1</td>
<td>Office XP SDK</td>
</tr>
<tr>
<td></td>
<td>f/w 4.6.8</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Microsoft Office Smart Tags SDK (Optional)

- Office 2003 SDK
- Office XP SDK
NOTES: Microsoft Authenticode Integration is tested with Luna Clients in FIPS and HA mode also.

Prerequisites

Configuring SafeNet Luna Network HSM 7.x

SafeNet Luna Network HSM allows to create Per-Partition Security Officer (PPSO) partition. HSM Administrator is not Security Officer (SO) for PPSO partitions. The HSM SO/Administrator elects to create a partition as PPSO-type, which creates an empty structure that is handed to the new owner, who initializes the partition to create the Partition Security Officer (PSO) role or identity for management functions. The PSO in turn creates the partition Crypto Officer (CO) to control client cryptographic operations on the partition.

Refer to the SafeNet Luna HSM documentation for installation steps and details regarding the configuration and setup of the box on UNIX/Windows systems. Before you get started ensure the following:

- SafeNet Luna Network HSM appliance and a secure admin password.
- SafeNet Luna Network HSM, and a hostname, suitable for your network.
- SafeNet Luna Network HSM network parameters are set to work with your network.
- Initialize the HSM on the SafeNet Luna Network HSM appliance.
- Create and exchange certificates between the SafeNet Luna Network HSM and your Client system.
- Create a partition on the HSM that will be later used by Microsoft Authenticode.
- Register the Client with the partition. And run the "vtl verify" command on the client system to display a partition from SafeNet Luna HSM. The general form of command is "C:\Program Files\SafeNet\LunaClient> vtl verify" for Windows and "/usr/safenet/lunaclient/bin/vtl verify" for Unix.
- Initialize the Partition as mentioned in steps below for Password/PED based respectively.
- Enabled Partition "Activation" and "Auto Activation" (Partition policy settings 22 and 23 (applies to SafeNet Luna Network HSM with Trusted Path Authentication [which is FIPS 140-2 level 3] only).

Initialize the Partition SO and Crypto Officer Roles on a PW-Auth Partition

These instructions assume a password-authenticated SafeNet Luna Network HSM that has been initialized, and an application partition has been created, capable of having its own Security Officer.

- **Initialize the Partition SO role**

  Set the active slot to the created, uninitialized, application partition.
  Type `slot set -slot <slot number>`
  ```
  lunacm:> slot set -slot 0
  Current Slot Id:    0     (Luna User Slot 7.0.0 (Password) Signing With Cloning Mode)
  Command Result : No Error
  ```
Initialize the application partition, to create the partition’s Security Officer (SO).
Type **partition init -label** `<partition label>`
lunacm:> par init -label `<part_label>` -password `<part_password>`
   You are about to initialize the partition.
   All partition objects will be destroyed.
   Are you sure you wish to continue?
   Type 'proceed' to continue, or 'quit' to quit now -> proceed
   Command Result: No Error

- **Initialize the Crypto Officer role**
  a. The SO of the application partition can now assign the first operational role within the new partition.
     Type **role login -name** `Partition SO`.
        lunacm:> role login -name Partition SO
  b. Type **role init -name** `Crypto Officer`.
        lunacm:> role init -name Crypto Officer
  c. The application partition SO can create the Crypto Officer, but only the Crypto Officer can create the
     Crypto User. Therefore, the SO must log out to allow the Crypto Officer to log in.
     Type **role logout**.
        lunacm:> role logout

**Initialize the Partition SO and Crypto Officer Roles on a PED-Auth Partition**

These instructions assume a PED-authenticated SafeNet Luna Network HSM that has been initialized, and an
application partition has been created, capable of having its own Security Officer.

Take the following steps to initialize the PSO and CO roles:

- **Initialize the Partition SO role**
  Set the active slot to the created, uninitialized, application partition.
  Type **slot set -slot** `<slot number>`
  lunacm:> slot set -slot 0
  Current Slot Id: 0 (Luna User Slot 7.0.0 (PED) Signing With Cloning Mode)
  Command Result : No Error
  Initialize the application partition, to create the partition’s Security Officer (SO).
  Type **partition init -label** `<partition label>`
  lunacm:> par init -label `<part_label>`
     You are about to initialize the partition.
     All partition objects will be destroyed.
     Are you sure you wish to continue?
     Type 'proceed' to continue, or 'quit' to quit now -> proceed
     Please attend to the PED.
  Respond to SafeNet PED prompts...
  Command Result : No Error

- **Initialize the Crypto Officer role**
  The SO of the application partition can now assign the first operational role within the new partition.
  Type **role login -name** `Partition SO`.
  Type **role init -name** `Crypto Officer`.
  lunacm:> role init -name Crypto Officer
  Please attend to the PED.
Respond to SafeNet PED prompts...
Command Result: No Error

The application partition SO can create the Crypto Officer, but only the Crypto Officer can create the Crypto User. Therefore, the SO must log out to allow the Crypto Officer to log in.

Type `role logout`.

Now, the Crypto Officer, or an application using the CO’s challenge secret/password can perform cryptographic operations in the partition, as soon as the Crypto Officer logs in with `role login -name Crypto Officer`. However, the Crypto Officer can create, modify and delete crypto objects within the partition, in addition to merely using existing crypto objects (sign/verify). You can also create a limited-capability role called Crypto User that can use the objects created by the Crypto Officer, but cannot modify them.

NOTE: The black Crypto Officer PED key/Crypto Officer password is valid for the initial login only. You must change the initial credential on the key using the command `role changepw` during the initial login session, or a subsequent login. Failing to change the credential will result in a CKR_PIN_EXPIRED error while performing role-dependent actions.

Configuring SafeNet Luna Network HSM (v4.x/5.x/6.x)

Refer to the SafeNet Luna HSM documentation for installation steps and details regarding the configuration and setup of the box on UNIX systems. Before you get started ensure the following:

- SafeNet Luna Network HSM appliance and a secure admin password.
- SafeNet Luna Network HSM, and a hostname, suitable for your network.
- SafeNet Luna Network HSM network parameters are set to work with your network.
- Initialize the HSM on the SafeNet Luna Network HSM appliance.
- Create and exchange certificates between the SafeNet Luna Network HSM and your Client system.
- Create a partition on the HSM, remember the partition password that will be later used by Microsoft Authenticode.
- Register the Client with the partition. And run the "vtl verify" command on the client system to display a partition from SafeNet Luna Network HSM. The general form of command is "C:\Program Files\SafeNet\LunaClient> vtl verify" for Windows and "/usr/safenet/lunaclient/bin/vtl verify" for Unix.
- Enabled Partition "Activation" and "Auto Activation" (Partition policy settings 22 and 23 (applies to SafeNet Luna Network HSM with Trusted Path Authentication [which is FIPS 140-2 level 3] only).

SafeNet Luna PCI-E HSM Setup

Refer to the SafeNet Luna PCI documentation for installation steps and details regarding configuring and setting up the box on Windows systems. Before you get started, ensure the following:

- Initialize the HSM on the SafeNet Luna PCI appliance.
- Create a partition on the HSM that will be later used by Microsoft Authenticode.
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).

**Using Luna 6.x/7.x 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 Luna 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 SafeNet Luna HSM is in FIPS mode.

NOTE: The above configuration is valid for Luna 7.x and Luna 6.x (FW Version 6.22.0 and above only).

**Microsoft Authenticode Setup**

**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*.

**Installing the Office Smart Tags SDK**

In order to demonstrate the Authenticode technology the Microsoft Office Smart Tags SDK is required.

Refer to the *Microsoft Office Smart Tags SDK installation documentation*.

NOTE: Any names used in the commands for the configuration of Authenticode that contain spaces must be in double quotes.
Perform the following steps to integrate the Safenet Luna HSM with Microsoft Authenticode:

1. In order to integrate the Safenet Luna HSM with Microsoft Authenticode, the Luna CSP "Luna Cryptographic Services for Microsoft Windows" must be used with the makecert command:

   makecert -sk noi1-501706key -sp "Luna Cryptographic Services for Microsoft Windows" -r -n CN=noi1-501706 -ss TestStore noi1-501706.cer

   where:

   -sk The location of the subject's key container which holds the private key.
   -sp Subject CryptoAPI’s provider name.
   -r Create a self –signed certificate.
   -n The name of the publisher’s certificate.
   -ss The name of the subject’s certificate store in which the generated certificate will be stored.

2. Create a Software Publishing Certificate (SPC) from the generated certificate.

   • Traverse to the Microsoft Platform SDK bin directory, i.e. "C:\Program Files\Microsoft Platform SDK\Bin"
   
   Cert2Spc noi1-501706.cer noi1-501706.spc

3. Sign and Time Stamp the code. The following steps illustrate how to Sign and Time Stamp the code using signtool GUI:

   • Open the command prompt and traverse to the "Microsoft Platform SDK bin" directory, i.e. "C:\Program Files\Microsoft Platform SDK\Bin"
   
   • Run the command "signtool signwizard" to launch the digital signing wizard.
   
   • Click Next on the Welcome screen.
   
   • Select the desired file (.dll, .exe etc.) to sign. Click Next.
     
     e.g. C:\Program Files\Smart Tag SDK\Simple VB Sample\SimpleTerm.dll installed with the Smart Tag SDK.

   • Choose the Custom option in the Signing Options window and click Next.

   • In the Signature Certificate Window, Click Select from File. and browse to the generated Software Publishing Certificate (.spc file). Click Next.
• Select Private Key in a CSP if key is generated using Cryptographic Service Provider i.e. "Luna Cryptographic Services for Microsoft Windows", select the appropriate CSP, Key container, click Next.

• Select Desired Hash Algorithm and Click Next.

• Click Next on the Additional Certificates window.

• Optionally add a description on Data Description window and Click Next.

• Select Add a timestamp to the data and give the time stamping URL.

• The Complete the Digital Signature Wizard window displays. Click Finish.

• The Digital Signing Wizard was completed successfully information window displays. Click OK. Microsoft Authenticode integration with SafeNet Luna HSM is completed.
Integrating SafeNet Luna HSM with Microsoft Authenticode (Windows SDK for Windows Server 2008)

Perform the following steps to integrate the SafeNet Luna HSM with Microsoft Authenticode:

1. Install Luna Cryptographic Service Provider (CSP) on Windows Server 2008.
   - Run the command `register.exe` to register Luna CSP. The general form of command is:
     `<Luna Client Installation Directory>CSP>register.exe`

   ```
   ********************************************
   *                                      *
   * Safenet Inc. LunaCSP, Partition Registration *
   * Protect the HSM’s challenge for the selected partitions. *
   * NOTE: *
   * This is a WEAK protection of the challenge!! *
   * After you have configured all applications that will use *
   * the LunaCSP, and ran them once, you MUST run: *
   * register /partition /strongprotect *
   * to strongly protect the registered challenges!! *
   ********************************************
   
   This procedure is a destructive procedure and will completely replace any previous settings!!
   Do you wish to continue?: [y/n] y
   
   Do you want to register the partition named 'part2'?[y/n]: y
   
   Enter challenge for partition 'part2':******
   
   Success registering the ENCRYPTED challenge for partition 'part2:1'.
   Only the LunaCSP will be able to use this data!

   Registered 1 partition(s) for use by the LunaCSP!
   ```
To register the Luna Cryptographic Services for Microsoft Windows. The general form of command is given below:

```
C:\Program Files (x86)\SafeNet\CSP>register.exe /l
```

Success registering SOFTWARE\Microsoft\Cryptography\Defaults\Provider\Luna Cryptographic Services for Microsoft Windows
Success registering SOFTWARE\Microsoft\Cryptography\Defaults\Provider\Luna SChannel Cryptographic Services for Microsoft Windows

2. In order to integrate the SafeNet Luna HSM with Microsoft Authenticode, the Luna CSP "Luna Cryptographic Services for Microsoft Windows" must be used with the `makecert` command:

Example:

```
C:\>makecert -sk mykey -sp "Luna Cryptographic Services for Microsoft Windows" -n CN=WIN2008-TEST1 -ss mycert 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 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.

Example:

```
C:\>cert2spc Test.cer Test.spc
```

4. Signing and Time Stamping the code. The following steps illustrate how to Sign and Time Stamp the code using `signtool GUI`:

- Open the command prompt and traverse to the Microsoft Platform SDK bin directory, i.e. `C:\Program Files\Microsoft SDKs\Windows\v6.1\Bin`
- Run the command `signtool signwizard` to launch the digital signing wizard.
- Click Next on the Welcome screen.
- Select the desired file (.dll, .exe etc.,) to sign and click Next.
  Example:
  `C:\Program Files\Microsoft Office 2003 Developer Resources\Microsoft Office 2003 Smart Tag SDK\Samples\Visual Basic 6.0 Sample\SimpleTerm\SimpleTerm.dll`
- Choose the Custom option in the Signing Options window and click Next.
- In the Signature Certificate Window, Click Select from File... and browse to the generated Software Publishing Certificate (.spc file) and click Next.
• Select **Private Key in a CSP** if key is generated using Cryptographic Service Provider (CSP), select the appropriate **CSP** i.e. **Luna Cryptographic Services for Microsoft Windows, Key container** and click **Next**.

• Select the desired **Hash Algorithm** and Click **Next**.

• Click **Next** on the Additional Certificates window.

• Optionally add a description on the **Data Description** window and Click **Next**.

• Select **Add a timestamp to the data** and give the time stamping URL. Click **Next**.

• The **Complete the Digital Signature Wizard** window displays. Click Finish.

• The **Digital Signing Wizard was completed successfully** window displays. Click OK.

You can use `signtool` command without GUI using the following command:

```
C:\> signtool sign /v /s storename /csp "Cryptographic Service Provider Name" /kc "KeyContainerName" /t timestamp URL "File to be signed"
```

```
C:\>"Program Files\Microsoft SDKs\Windows\v5.1\Bin\signtool.exe" sign /v /s mycert /csp "Luna Cryptographic Services for Microsoft Windows" /kc mykey /t http:/
timestamp.globalsign.com/scripts/timestamp.dll "Program Files (x86)\Microsoft Office 2003 Developer Resources\Microsoft Office 2003 Smart Tag SDK\Samples\Visual Basic 6.0 Sample\SimpleTerm\SimpleTerm.dll"
The following certificate was selected:
  Issued to: UNV2000-TEST1
  Issued by: Root Agency
  Expires: 1/1/2040 5:29:59 AM
  SHA1 hash: 3DAB5CF6D28AE60691B942E8282E2140C8EAD
Done Adding Additional Store

Attempting to sign: \Program Files (x86)\Microsoft Office 2003 Developer Resources\Microsoft Office 2003 Smart Tag SDK\Samples\Visual Basic 6.0 Sample\SimpleTerm\SimpleTerm.dll
Successfully signed and timestamped: \Program Files (x86)\Microsoft Office 2003 Developer Resources\Microsoft Office 2003 Smart Tag SDK\Samples\Visual Basic 6.0 Sample\SimpleTerm\SimpleTerm.dll

Number of files successfully Signed: 1
Number of warnings: 0
Number of errors: 0
```

Microsoft Authenticode integration with SafeNet Luna HSM is completed.
Integrating SafeNet Luna HSM with Microsoft Authenticode (Windows SDK for Windows Server 2012 R2/2016)

Perform the following steps to integrate the SafeNet Luna HSM with Microsoft Authenticode:

   - Run the command, `register.exe` to register Luna CSP. The general form of command is given below:
     ```
     <Luna Client Installation Directory>CSP>register.exe
     ```
   - 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. In order to integrate the SafeNet Luna HSM with Microsoft Authenticode, the Luna CSP "Luna Cryptographic Services for Microsoft Windows" must be used with the `makecert` command:

   ```
   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
   ```
4. Signing and Time Stamping the code.

You can Sign and Time Stamp the code using signtool as follows:

```
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.

Microsoft Authenticode integration with SafeNet Luna HSM is completed and ready for use.

Microsoft Authenticode integration with SafeNet Luna HSM is completed and ready for use.
Integrating SafeNet Luna HSM with MS Strong Name (Windows Server 2012 R2)

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. Strong names prevent spoofing of the code by a third party (that is, of course, as long as you keep the private key secure).

You can secure the strong name private key on SafeNet Luna HSM.

1. Configure Luna Cryptographic Service Provider (CSP) on Windows Server.
   - 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
     ```
   - 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. In order to integrate the SafeNet Luna HSM with Microsoft Strong Name, the Luna CSP "Luna Cryptographic Services for Microsoft Windows" must be used with the `makecert` command, open the command prompt and run the following command:
   ```
   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 as a 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.
5. To sign a .NET assembly, write any C# program and open the Visual Studio command prompt to compile
the program and 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.

6. Sign the generated exe with Strong Name using the following command:
sn.exe -Rc C:\Users\Administrator\Desktop\myapp.exe mykey
where "mykey" is the key container in which you have generated the key-pair.
7. Verify the assembly is Strong Name signed using the following command:

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

You have successfully signed and verified a .Net assembly using the key-pair generated on SafeNet Luna HSM with Strong Name.
Integrating SafeNet Luna HSM with MS Mage/ClickOnce (Windows Server 2012 R2)

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 Mage/ClickOnce signing, refer Microsoft Documentation.

SafeNet Luna 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.

1. Install Luna Cryptographic Service Provider (CSP) on Windows Server.
   - 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`
   - 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. In order to integrate the SafeNet Luna HSM with Microsoft Mage\ClickOnce Signing, the Luna CSP "Luna Cryptographic Services for Microsoft Windows" must be used with the makecert command, open the command prompt and run the following command:
   ```
   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.
NOTE: After generating the certificate, you can use this certificate in Visual Studio to sign the Application/Deployment manifest. You need to 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 you have generated in step 1.

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

1. Create a directory where you will store your ClickOnce deployment files.
2. In the deployment directory just created, create a version subdirectory. If this is the first time that you are deploying the application, 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, you can 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.
NOTE: Make sure to include the dot (.) after the -FromDirectory option, which indicates the current directory. If you do not include the dot, you must specify the path to your application files.

6. Sign the application manifest with your 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 you have generated the certificate using the makecert command.

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 you use the default, and
users will install your 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 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.
Integrating SafeNet Luna HSM with Microsoft HCK (Windows Server 2012)

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 Luna HSM is used to secure the signing keys so that your signing keys never access by any unauthorized entity. Microsoft HCK uses the RSA keys for signing the packages.

Microsoft HCK is a 32 bit application so you have to use the Luna Clients with 32 bit CSP.

1. Install Luna Cryptographic Service Provider (CSP) on Windows Server.
   - 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
     ```
   - 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 /I
     ```

2. As this is a 32 bit setup, to verify the registered cryptographic providers, browse to "C:\Windows\SysWOW64" and execute "certutil \-csplist"

3. In order to integrate the SafeNet Luna 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 method that you can use to generate the signing certificate:

**Method 1**

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 –repairstore" utility.
8. Open the certificate.
10. Select the **Serial number** field.
11. Copy the serial number or thumb print.
12. Execute the "certutil -repairstore -user My "SerialNumber or ThumbPrint" command from the SysWOW64 directory to map the private key (on the HSM) with the certificate.
13. After the `repairsotre` command has been successfully executed, refresh the certificate manager snap in, open the certificate and confirm the message at the bottom is displayed.

![Certificate Information]

**Method 2**

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
```

   - `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. Browse through the various tabs to check if the project imported is correct.
## Integrating SafeNet Luna HSM with Microsoft HCK (Windows Server 2012)

### Windows Hardware Certification Kit

#### W81x64ClientOnAccess

<table>
<thead>
<tr>
<th>Status</th>
<th>Test Name</th>
<th>Type</th>
<th>Length</th>
<th>Target</th>
<th>Machine(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Anonymous Pipe</td>
<td></td>
<td>30m</td>
<td>SAVONACC</td>
<td>SAVHCK-W1</td>
</tr>
<tr>
<td>✔️</td>
<td>Antivirus Installable</td>
<td></td>
<td>30m</td>
<td>SAVONACC</td>
<td>SAVHCK-W1</td>
</tr>
<tr>
<td>✔️</td>
<td>File IO Tests</td>
<td>05h 00m</td>
<td>SAVONACC</td>
<td>SAVHCK-W1</td>
<td></td>
</tr>
<tr>
<td>✔️</td>
<td>Installable File Sys</td>
<td>02h 00m</td>
<td>SAVONACC</td>
<td>SAVHCK-W1</td>
<td></td>
</tr>
<tr>
<td>✔️</td>
<td>Mailslot Basic</td>
<td>30m</td>
<td>SAVONACC</td>
<td>SAVHCK-W1</td>
<td></td>
</tr>
<tr>
<td>✔️</td>
<td>Mapped File IO</td>
<td>30m</td>
<td>SAVONACC</td>
<td>SAVHCK-W1</td>
<td></td>
</tr>
<tr>
<td>✔️</td>
<td>Named Pipe Basic</td>
<td>30m</td>
<td>SAVONACC</td>
<td>SAVHCK-W1</td>
<td></td>
</tr>
<tr>
<td>✔️</td>
<td>Named Pipe Kernel</td>
<td>30m</td>
<td>SAVONACC</td>
<td>SAVHCK-W1</td>
<td></td>
</tr>
<tr>
<td>✔️</td>
<td>Named Pipe MSRC</td>
<td>30m</td>
<td>SAVONACC</td>
<td>SAVHCK-W1</td>
<td></td>
</tr>
<tr>
<td>✔️</td>
<td>Named Pipe Reject</td>
<td>30m</td>
<td>SAVONACC</td>
<td>SAVHCK-W1</td>
<td></td>
</tr>
<tr>
<td>✔️</td>
<td>Named Pipe State</td>
<td>30m</td>
<td>SAVONACC</td>
<td>SAVHCK-W1</td>
<td></td>
</tr>
<tr>
<td>✔️</td>
<td>Object ID test</td>
<td>30m</td>
<td>SAVONACC</td>
<td>SAVHCK-W1</td>
<td></td>
</tr>
<tr>
<td>✔️</td>
<td>Oplocks Test</td>
<td>09h 00m</td>
<td>SAVONACC</td>
<td>SAVHCK-W1</td>
<td></td>
</tr>
<tr>
<td>✔️</td>
<td>Registry Callback Test</td>
<td>30m</td>
<td>SAVONACC</td>
<td>SAVHCK-W1</td>
<td></td>
</tr>
<tr>
<td>✔️</td>
<td>ReparsePoints</td>
<td>01h 00m</td>
<td>SAVONACC</td>
<td>SAVHCK-W1</td>
<td></td>
</tr>
<tr>
<td>✔️</td>
<td>Sleep and PNP (disabled)</td>
<td>01h 30m</td>
<td>SAVONACC</td>
<td>SAVHCK-W1</td>
<td></td>
</tr>
<tr>
<td>✔️</td>
<td>Syscache Test</td>
<td>03h 00m</td>
<td>SAVONACC</td>
<td>SAVHCK-W1</td>
<td></td>
</tr>
<tr>
<td>✔️</td>
<td>TDI filters and LSPs</td>
<td>01m</td>
<td>SAVONACC</td>
<td>SAVHCK-W1</td>
<td></td>
</tr>
</tbody>
</table>

### Windows Hardware Certification Kit

#### W81x64ClientOnAccess

<table>
<thead>
<tr>
<th>Status</th>
<th>Test Name</th>
<th>Target</th>
<th>Machine(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>✔️</td>
<td>File IO Tests</td>
<td>SAVONACC</td>
<td>SAVHCK-W81X1</td>
</tr>
<tr>
<td>✔️</td>
<td>Installable File System Filter Test</td>
<td>SAVONACC</td>
<td>SAVHCK-W81X1</td>
</tr>
<tr>
<td>✔️</td>
<td>Mailslot Basic</td>
<td>SAVONACC</td>
<td>SAVHCK-W81X1</td>
</tr>
<tr>
<td>✔️</td>
<td>Mapped File IO</td>
<td>SAVONACC</td>
<td>SAVHCK-W81X1</td>
</tr>
<tr>
<td>✔️</td>
<td>Named Pipe Basic</td>
<td>SAVONACC</td>
<td>SAVHCK-W81X1</td>
</tr>
<tr>
<td>✔️</td>
<td>Named Pipe Kernel Security</td>
<td>SAVONACC</td>
<td>SAVHCK-W81X1</td>
</tr>
<tr>
<td>✔️</td>
<td>Named Pipe MSRC249</td>
<td>SAVONACC</td>
<td>SAVHCK-W81X1</td>
</tr>
<tr>
<td>✔️</td>
<td>Named Pipe Reject Remeta Clients</td>
<td>SAVONACC</td>
<td>SAVHCK-W81X1</td>
</tr>
<tr>
<td>✔️</td>
<td>Named Pipe State</td>
<td>SAVONACC</td>
<td>SAVHCK-W81X1</td>
</tr>
<tr>
<td>✔️</td>
<td>Object ID test</td>
<td>SAVONACC</td>
<td>SAVHCK-W81X1</td>
</tr>
<tr>
<td>✔️</td>
<td>Oplocks Test</td>
<td>SAVONACC</td>
<td>SAVHCK-W81X1</td>
</tr>
<tr>
<td>✔️</td>
<td>Registry Callback Tests</td>
<td>SAVONACC</td>
<td>SAVHCK-W81X1</td>
</tr>
<tr>
<td>✔️</td>
<td>ReparsePoints</td>
<td>SAVONACC</td>
<td>SAVHCK-W81X1</td>
</tr>
<tr>
<td>✔️</td>
<td>Sleep and PNP (disabled)</td>
<td>SAVONACC</td>
<td>SAVHCK-W81X1</td>
</tr>
<tr>
<td>✔️</td>
<td>Syscache Test</td>
<td>SAVONACC</td>
<td>SAVHCK-W81X1</td>
</tr>
<tr>
<td>✔️</td>
<td>TDI filters and LSPs</td>
<td>SAVONACC</td>
<td>SAVHCK-W81X1</td>
</tr>
<tr>
<td>✔️</td>
<td>Winsock Core Functional Test</td>
<td>SAVONACC</td>
<td>SAVHCK-W81X1</td>
</tr>
</tbody>
</table>
6. After verification, go to the package tab and click on create package to sign the package. It will ask 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. At 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 Tips

Problem

If you are operating HSM in FIPS Mode and you are facing the following problem while running the Makecert command to generate the certificate:

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:

<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.