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.

© 2010-18 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 Number: 007-011108-001, Rev. AG
Release Date: June 2018
Preface .......................................................................................................................... 5
Scope ............................................................................................................................. 5
Document Conventions ............................................................................................... 5
Command Syntax and Typeface Conventions ............................................................... 6
Support Contacts ......................................................................................................... 7
1 Introduction ............................................................................................................... 8
Overview ...................................................................................................................... 8
3rd Party Application Details ...................................................................................... 8
Support for Higher Length Asymmetric Keys ............................................................... 9
Supported Platforms .................................................................................................. 10
SafeNet Luna HSM (v7.x) ......................................................................................... 10
SafeNet Luna HSM (v4.x/5.x/6.x) ............................................................................. 11
Prerequisites ............................................................................................................... 14
Configuring SafeNet Luna Network HSM 7.x ............................................................ 14
Configuring SafeNet Luna Network HSM (v4.x/5.x/6.x) ........................................... 16
Using Luna 6.x in FIPS Mode .................................................................................... 17
SafeNet Luna PCI-E HSM Setup .............................................................................. 17
SafeNet Luna HSM Configuration Settings .............................................................. 17
Luna EKM Setup ........................................................................................................ 18
SQL Server Setup ...................................................................................................... 18
2 SafeNet Luna HSM Integration with SQL Server .................................................... 19
Setting up SafeNet Luna HSM with SQL Server ....................................................... 19
Enabling EKM Provider option .................................................................................. 19
Register Luna EKM Provider ..................................................................................... 19
Setting up Credential for Luna EKM Provider ......................................................... 20
Using Luna EKM Provider ......................................................................................... 21
Migration from SQL EKM to Luna EKM ................................................................. 26
Using Extensible Key Management on a SQL Server Failover Cluster ...................... 28
3 SafeNet Luna HSM Integration with SQL Server High Availability (Always On) Group .. 30
SQL Server Setup ...................................................................................................... 30
Enabling EKM Provider Option ................................................................................. 30
Registering Luna EKM Provider ............................................................................... 31
Setting up Credential for Luna EKM Provider ......................................................... 31
Creating the Always On Availability Group ............................................................. 32
Creating the Encryption Keys for Availability Group Database ............................... 33
Add the encrypted database in to the availability group ......................................... 38
4 SafeNet Luna HSM Integration with SQL Server Always Encrypted .......................... 40
   Overview.................................................................................................................................. 40
   Setting up SQL Server Always Encrypted using SafeNet Luna HSM .................................... 41
      Configure SafeNet KSP........................................................................................................ 41
      Generating Column Master Key ........................................................................................... 41
      Generating Column Encryption Key ...................................................................................... 44
      Implementing Always Encrypted ............................................................................................ 47
      Viewing Data with Always Encrypted Implemented in SQL SSMS ................................. 50

5 Troubleshooting Tips............................................................................................................... 52
Preface

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

Scope

This guide provides instructions for setting up a small test lab with Microsoft SQL Server running with SafeNet Luna HSM for securing the Master Keys. It explains how to install and configure the software required for setting up Microsoft SQL Server while storing master key on SafeNet Luna 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>
</table>
| Address                     | Gemalto  
4690 Millennium Drive  
Belcamp, Maryland  21017, USA                                               |
| Phone                       | US  
1-800-545-6608                                                                  |
|                             | International  
1-410-931-7520                                                                  |
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. |
Introduction

Overview

SQL Server enables use of HSM devices for storage of keys and cryptographic operations such as key creation, deletion, encryption, decryption etc. by using the Extensible Key Management (EKM) feature. This is a more secure solution because the encryption keys do not reside with encryption data. Data can be encrypted by using encryption keys that are accessible only by the database users on the external EKM/HSM module. SafeNet provides Luna EKM, which includes the EKM Provider Library for SafeNet Luna HSM that can be used to setup Extensible Key Management (EKM) for SQL Server and facilitates the integration with SafeNet Luna HSM.

This document provides low-level details of how the SafeNet Luna Hardware Security Modules (HSM) can be made to work with SQL Server. You must have basic knowledge of using SQL Server and HSM concepts to make full use of the recommendations in this document. This document is intended for:

- Developers and enterprise IT professionals who are planning or implementing a HSM deployment. This includes IT security administrators and IT personnel.

The installation is performed in several steps:

- Install and configure SafeNet Luna HSM.
- Install and configure SafeNet Luna EKM.

3rd Party Application Details

Microsoft SQL Server is a database platform for large-scale online transaction processing (OLTP), data warehousing, and e-commerce applications. It is also a business intelligence platform for data integration, analysis, and reporting solutions.
The following diagram shows the relationships between the database master key and Hardware Security Modules:

![Diagram showing relationships between database master key and Hardware Security Modules]

**Support for Higher Length Asymmetric Keys**

We have included support for creating higher length asymmetric keys: RSA_3072 and RSA_4096 from Luna EKM v1.3 onwards. However during our integration testing, we identified an issue in TDE when encrypting the DEK using RSA_4096 key. This issue has been reported to Microsoft technical support and we are awaiting a resolution. At this time, we recommend to use a maximum key length of RSA_3072 for the TDE. We will retest and update the integration guide when Microsoft resolves this issue.

*NOTE:* For Luna EKM setup, contact Customer support.

DOC ID for EKM 1.3 is **KB0014957** and for EKM 1.4 is **KB0016274**.
## Supported Platforms

### SafeNet Luna HSM (v7.x)

**Microsoft® SQL Server® 2016 SP1 (Always Encrypted)**

<table>
<thead>
<tr>
<th>Platforms Tested</th>
<th>SafeNet Luna Network HSM Appliance Version</th>
<th>Firmware</th>
<th>Luna Client Software Version</th>
<th>SafeNet KSP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows Server 2016</td>
<td>7.2.0</td>
<td>7.2.0</td>
<td>7.2.0</td>
<td>32-bit</td>
</tr>
<tr>
<td>Windows Server 2016</td>
<td>7.1.0</td>
<td>7.1.0</td>
<td>7.1.0</td>
<td>32-bit</td>
</tr>
</tbody>
</table>

**NOTE:** Always Encrypted feature is tested in FIPS and Non-FIPS mode.

### Microsoft® SQL Server® 2016 SP1

<table>
<thead>
<tr>
<th>Platforms Tested</th>
<th>SafeNet Luna Network HSM Appliance Version</th>
<th>Firmware</th>
<th>Luna Client Software Version</th>
<th>EKM Software Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows Server 2016</td>
<td>7.2.0</td>
<td>7.2.0</td>
<td>7.2.0</td>
<td>EKM v1.4</td>
</tr>
<tr>
<td>Windows Server 2016</td>
<td>7.1.0</td>
<td>7.1.0</td>
<td>7.1.0</td>
<td>EKM v1.4</td>
</tr>
</tbody>
</table>
| Windows Server 2012 R2   | 7.0.0                                      | 7.0.1    | 7.0.0                         | EKM v1.4
| Windows Server 2012 R2   |                                            |          |                               | EKM v1.3             |
## SafeNet Luna HSM (v4.x/5.x/6.x)

### Microsoft® SQL Server® 2016 SP1

<table>
<thead>
<tr>
<th>Platforms Tested</th>
<th>SafeNet Luna HSM Client Software Version</th>
<th>SafeNet Luna HSM Appliance Software Version</th>
<th>Appliance Firmware Version</th>
<th>EKM Software Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows Server 2016</td>
<td>6.3.0</td>
<td>v6.3.0</td>
<td>6.10.9 and 6.27.0</td>
<td>EKM v1.4 EKM v1.3</td>
</tr>
<tr>
<td>Windows Server 2012 R2</td>
<td>6.2.2</td>
<td>v6.2.2</td>
<td>6.24.3</td>
<td>EKM v1.3</td>
</tr>
</tbody>
</table>

### Microsoft® SQL Server® 2016 CTP3

<table>
<thead>
<tr>
<th>Platforms Tested</th>
<th>SafeNet Luna HSM Client Software Version</th>
<th>SafeNet Luna HSM Appliance Software Version</th>
<th>Appliance Firmware Version</th>
<th>EKM Software Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows Server 2012 R2</td>
<td>6.2.2</td>
<td>v6.2.2</td>
<td>6.10.9 and 6.24.3</td>
<td>EKM v1.2</td>
</tr>
<tr>
<td>Windows Server 2012 R2</td>
<td>6.2</td>
<td>v6.2.1</td>
<td>6.10.9 and 6.24.2</td>
<td>EKM v1.2</td>
</tr>
<tr>
<td>Windows Server 2012 R2</td>
<td>6.2</td>
<td>v6.2.0</td>
<td>6.10.9</td>
<td>EKM v1.2</td>
</tr>
<tr>
<td>Windows Server 2012 R2</td>
<td>6.1</td>
<td>v6.1.0</td>
<td>6.10.9</td>
<td>EKM v1.2</td>
</tr>
</tbody>
</table>

### Microsoft® SQL Server® 2014

<table>
<thead>
<tr>
<th>Platforms Tested</th>
<th>Luna Client Software Version</th>
<th>SafeNet Luna HSM Appliance Software Version</th>
<th>Appliance Firmware Version</th>
<th>EKM Software Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows Server 2012 R2</td>
<td>Luna Client 6.1</td>
<td>Luna SA v6.1.0</td>
<td>6.23.0</td>
<td>EKM v1.2</td>
</tr>
<tr>
<td></td>
<td>Luna Client 5.4.1</td>
<td>Luna SA v6.0</td>
<td>6.22.0</td>
<td>EKM v1.1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Luna SA 5.4.1</td>
<td>6.21.0</td>
<td>EKM v1.1</td>
</tr>
<tr>
<td>Windows Server 2008 R2 SP1</td>
<td>Luna Client 5.4.1</td>
<td>Luna SA v5.4.1</td>
<td>6.21.0</td>
<td>EKM v1.1</td>
</tr>
</tbody>
</table>
### Microsoft® SQL Server® 2014 CTP2

<table>
<thead>
<tr>
<th>Platforms Tested</th>
<th>Luna Client Software Version</th>
<th>SafeNet Luna HSM Appliance Software Version</th>
<th>Appliance Firmware Version</th>
<th>EKM Software Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows Server 2012 Standard</td>
<td>Luna Client 5.2.1</td>
<td>Luna SA v5.2.1</td>
<td>6.10.1</td>
<td>EKM v1.1</td>
</tr>
</tbody>
</table>

### Microsoft® SQL Server® 2014 CTP1

<table>
<thead>
<tr>
<th>Platforms Tested</th>
<th>Luna Client Software Version</th>
<th>SafeNet Luna HSM Appliance Software Version</th>
<th>Appliance Firmware Version</th>
<th>EKM Software Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows Server 2008 R2 SP1</td>
<td>Luna Client 5.2.1</td>
<td>Luna SA v5.2.1</td>
<td>6.10.1</td>
<td>EKM v1.1</td>
</tr>
</tbody>
</table>

### Microsoft® SQL Server® 2012

<table>
<thead>
<tr>
<th>Platforms Tested</th>
<th>Luna Client Software Version</th>
<th>SafeNet Luna HSM Appliance Software Version</th>
<th>Appliance Firmware Version</th>
<th>EKM Software Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows Server 2012 R2</td>
<td>Luna Client 6.2.1</td>
<td>V6.2.1</td>
<td>6.24.2</td>
<td>EKM v1.2</td>
</tr>
<tr>
<td>Windows Server 2012 R2</td>
<td>Luna Client 6.2.1</td>
<td>V6.2.1</td>
<td>6.10.9</td>
<td>EKM v1.2</td>
</tr>
<tr>
<td>Windows Server 2012 R2</td>
<td>Luna Client 5.4.1</td>
<td>Luna SA v5.4.1</td>
<td>6.21.0</td>
<td>EKM v1.1</td>
</tr>
<tr>
<td>Windows Server 2012 Standard</td>
<td>Luna Client 5.4.1</td>
<td>Luna SA v5.4.1</td>
<td>6.21.0</td>
<td>EKM v1.1</td>
</tr>
<tr>
<td></td>
<td>Luna Client 5.2.1</td>
<td>Luna SA v5.4.1</td>
<td>6.21.0</td>
<td>EKM v1.1</td>
</tr>
<tr>
<td>Windows Server 2008 R2</td>
<td>Luna Client 5.4.1</td>
<td>Luna SA v5.4.1</td>
<td>6.21.0</td>
<td>EKM v1.1</td>
</tr>
<tr>
<td>Windows Server 2008 R2</td>
<td>Luna Client 5.2.1</td>
<td>Luna SA v5.4.1</td>
<td>6.10.1</td>
<td>EKM v1.1</td>
</tr>
<tr>
<td>Windows Server 2008 R2</td>
<td>Luna SA CS v5.1.1</td>
<td>Luna SA v5.4.1</td>
<td>EKM v1.1</td>
<td></td>
</tr>
<tr>
<td>Windows Server 2008 R2</td>
<td>Luna SA CS v5.1</td>
<td>Luna SA v5.4.1</td>
<td>EKM v1.1</td>
<td></td>
</tr>
<tr>
<td>Windows Server 2008 R2</td>
<td>Luna PCI CS v5.0</td>
<td>Luna SA v5.4.1</td>
<td>EKM v1.1</td>
<td></td>
</tr>
<tr>
<td>Windows Server 2008 (32-bit)</td>
<td>Luna SA CS v5.1.1</td>
<td>Luna SA v5.4.1</td>
<td>EKM v1.0.2</td>
<td></td>
</tr>
<tr>
<td>Windows Server 2008 (32-bit)</td>
<td>Luna PCI CS v5.0</td>
<td>Luna SA v5.4.1</td>
<td>EKM v1.0.2</td>
<td></td>
</tr>
</tbody>
</table>
### Microsoft® SQL Server® 2008 R2 SP2

<table>
<thead>
<tr>
<th>Platforms Tested</th>
<th>Luna Client Software Version</th>
<th>SafeNet Luna HSM Appliance Software Version</th>
<th>Appliance Firmware Version</th>
<th>EKM Software Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows Server 2008 R2</td>
<td>Luna Client 5.4.1</td>
<td>Luna SA v5.4.1</td>
<td>6.21.0</td>
<td>EKM v1.1</td>
</tr>
</tbody>
</table>

### Microsoft® SQL Server® 2008 R2

<table>
<thead>
<tr>
<th>Platforms Tested</th>
<th>Luna Client Software Version</th>
<th>SafeNet Luna HSM Appliance Software Version</th>
<th>Appliance Firmware Version</th>
<th>EKM Software Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows Server 2012 Standard</td>
<td>Luna Client 5.4.1</td>
<td>Luna SA v5.4.1</td>
<td>6.21.0</td>
<td>EKM v1.1</td>
</tr>
<tr>
<td>Windows Server 2008 R2</td>
<td>Luna Client 5.4.1</td>
<td>Luna SA v5.4.1</td>
<td>6.21.0</td>
<td>EKM v1.1</td>
</tr>
<tr>
<td>Windows Server 2012 Standard</td>
<td>Luna Client 5.2.1</td>
<td>Luna SA v5.2.1</td>
<td>6.10.1</td>
<td>EKM v1.1</td>
</tr>
<tr>
<td>Windows Server 2008 R2</td>
<td>Luna Client 5.2.1</td>
<td>Luna SA v5.2.1</td>
<td>6.10.1</td>
<td>EKM v1.1</td>
</tr>
<tr>
<td>Windows Server 2008 R2</td>
<td>Luna SA CS v5.1</td>
<td>Luna SA v5.1</td>
<td>6.2.1</td>
<td>EKM v1.1</td>
</tr>
<tr>
<td>Windows Server 2008 R2</td>
<td>Luna SA CS v5.0</td>
<td>Luna SA v5.0</td>
<td>6.0.6</td>
<td>EKM v1.1</td>
</tr>
<tr>
<td>Windows Server 2008 R2</td>
<td>Luna SA CS v4.4.1</td>
<td>Luna SA v4.4.1</td>
<td>4.6.8</td>
<td>EKM v1.0.1</td>
</tr>
<tr>
<td>Windows Server 2008 R2</td>
<td>Luna PCI CS v5.0</td>
<td>Luna PCI v5.0</td>
<td>6.1.3</td>
<td>EKM v1.0.2</td>
</tr>
<tr>
<td>Windows Server 2008 R2</td>
<td>Luna PCI CS v3.0</td>
<td>Luna PCI v3.0</td>
<td>4.7.1</td>
<td>EKM v1.0.1</td>
</tr>
<tr>
<td>Windows Server 2008 (32-bit / 64-bit)</td>
<td>Luna SA CS v5.1</td>
<td>Luna SA v5.1</td>
<td>6.2.1</td>
<td>EKM v1.0.1</td>
</tr>
<tr>
<td>Windows Server 2008 (32-bit / 64-bit)</td>
<td>Luna SA CS v4.4.1</td>
<td>Luna SA v4.4.1</td>
<td>4.6.8</td>
<td>EKM v1.0.1</td>
</tr>
<tr>
<td>Windows Server 2003 R2 SP2 (32-bit)</td>
<td>Luna SA CS v5.0</td>
<td>Luna SA v5.0</td>
<td>6.0.6</td>
<td>EKM v1.0.1</td>
</tr>
<tr>
<td>Windows Server 2003 SP2 (32-bit / 64-bit)</td>
<td>Luna SA CS v4.4.1</td>
<td>Luna SA v4.4.1</td>
<td>4.6.8</td>
<td>EKM v1.0.1</td>
</tr>
</tbody>
</table>

### Microsoft® SQL Server® 2008 SP3

<table>
<thead>
<tr>
<th>Platforms Tested</th>
<th>Luna Client Software Version</th>
<th>SafeNet Luna HSM Appliance Software Version</th>
<th>Appliance Firmware Version</th>
<th>EKM Software Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows Server 2012 Standard</td>
<td>Luna Client 5.2.1</td>
<td>Luna SA v5.2.1</td>
<td>6.10.1</td>
<td>EKM v1.1</td>
</tr>
</tbody>
</table>
Platforms Tested | Luna Client Software Version | SafeNet Luna HSM Appliance Software Version | Appliance Firmware Version | EKM Software Version
--- | --- | --- | --- | ---
Windows Server 2008 R2 | Luna Client 5.4.1 | Luna SA v5.4.1 | 6.21.0 | EKM v1.1

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. This new owner is responsible for initializing 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 SQL server.
- 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 <part_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 <part_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`. 

Microsoft SQL Server Integration Guide
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 SQL server.
- 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).

---

**NOTE:** For PED based SafeNet Luna HSM make sure ProtectedAuthenticationPathFlagStatus is set to `1` in Misc Section of Chrystoki.conf file.
Using Luna 6.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 Network Luna HSM in FIPS mode, you have to make the following change in configuration file:

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

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

**NOTE:** The above configuration is valid for Luna 6.x (F/W Version 6.22.0).

**NOTE:** Firmware 6.10.9 supports SQL Server in the both FIPS and non-FIPS modes and firmware 6.24.x, 6.27.0 and 7.x supports SQL Server in the non-FIPS mode only.

SafeNet Luna PCI-E HSM Setup

Refer to the SafeNet 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 PCI HSM appliance.
- Create a partition on the HSM that will be later used by the SQL Server.
- Enable Partition "Activation" and "Auto Activation" (Partition policy settings 22 and 23 (applies to SafeNet PCI HSM with Trusted Path Authentication [which is FIPS 140-2 level 3] only).
- Use the following setting in Chrystoki Configuration file:

  ```plaintext
  [Misc]
  PE1746Enabled = 1
  ```

**NOTE:** The above configuration is valid for SafeNet Luna PCI 6.x (F/W Version 6.22.0 and above only).

SafeNet Luna HSM Configuration Settings

The Luna Client configuration file located at the following path needs to be changed for Luna v6.x:

```
C:\Program Files\SafeNet\LunaClient\crystoki.ini
```

This configuration file needs to be edited for slot id because by default it is set to 0. Set the slot id to 1 by making the following changes in the configuration file:

```plaintext
[Presentation]
OneBaseSlotId=1
```
Luna EKM Setup

A Windows-based installation program is provided to make the Luna EKM installation quick and easy.

LunaEKMConfig Utility

LunaEKM includes a command line configuration utility "LunaEKMConfig" that is used to register the Luna EKM. This command line utility gets installed in LunaEKM installation folder. It provides command to register slots, view slots, and to configure log settings.

Run the following commands provided in LunaEKMConfig.

1. `RegisterSlot`
   Register/Edit the Slot for the LunaEKM to use.

2. `ViewSlots`
   View List of the Slots/HSM configured with this client.

3. `LogSettings`
   Configure log settings for LunaEKM.
   
   - `LogLevel (NONE=0,INFO=1,DEBUG=2): <LogLevel>`
   - `LogFile name: <Name and location of LogFile>`

SQL Server Setup

SQL Server must be installed on the target machine to carry out the integration process. For a detailed installation procedure of SQL Server, refer to the Microsoft SQL Server online documentation.
SafeNet Luna HSM Integration with SQL Server

Setting up SafeNet Luna HSM with SQL Server

To perform SafeNet Luna HSM integration with SQL Server, Luna EKM software provides Luna EKM Provider in the form of EKM Library (i.e. LunaEKM.dll). The Luna EKM Provider can be used if the EKM Provider option is enabled in the SQL Server. This feature is available only on the Enterprise, Developer, and Evaluation editions of SQL Server. By default, Extensible Key Management is off.

Enabling EKM Provider option

To enable this feature, use the sp_configure command that has the following option and value, as in the following example:

To enable the Extensible Key Management option:

1. Open the SQL Server Management Studio.
2. Connect to the SQL Server.
3. Open a query window, and then run the following command:

```sql
sp_configure 'show advanced', 1
GO
RECONFIGURE
GO
sp_configure 'EKM provider enabled', 1
GO
RECONFIGURE
GO
```

**NOTE:** If sp_configure command used in editions other than Enterprise, Developer or Evaluation editions, an error is received.

Register Luna EKM Provider

To setup the Luna EKM provider, Luna EKM Software must be installed and needs to be registered with the SQL Server. Follow the below steps to create/register the provider:

To create/register the Luna EKM Provider:

1. Open the SQL Server Management Studio.
2. Connect to the SQL Server.

3. Open a query window, and then run the following command:
   
   ```
   CREATE CRYPTOGRAPHIC PROVIDER <Name of Cryptographic Provider>
   FROM FILE = '<Location of Luna EKM Provider Library>'
   ```
   
   where CRYPTOGRAPHIC PROVIDER can be any user defined unique name.

4. To view the list of EKM providers:
   
   ```
   SELECT [provider_id]
   ,[name]
   ,[guid]
   ,[version]
   ,[dll_path]
   ,[is_enabled]
   FROM [model].[sys].[cryptographic_providers]
   ```

5. To view the provider properties:
   
   ```
   SELECT [provider_id],[guid],[provider_version]
   ,[sqlcrypt_version]
   ,[friendly_name]
   ,[authentication_type]
   ,[symmetric_key_support]
   ,[symmetric_key_persistance]
   ,[symmetric_key_export]
   ,[symmetric_key_import]
   ,[asymmetric_key_support]
   ,[asymmetric_key_persistance]
   ,[asymmetric_key_export]
   ,[asymmetric_key_import]
   FROM [master].[sys].[dm_cryptographic_provider_properties]
   ```

---

### Setting up Credential for Luna EKM Provider

The next step is to create a CREDENTIAL for the Luna EKM Provider. Then the CREDENTIAL must be mapped to SQL User or Login to be able to use the Luna EKM Provider. A CREDENTIAL is basically used to access any external SQL Server resource such as SafeNet Luna HSM. Follow the below steps to create/map credential for the provider:

To create/map the CREDENTIAL for Luna EKM Provider:

1. Open a query window, and then run the following command:

   ```
   CREATE CREDENTIAL <Name of credential>
   WITH IDENTITY='<Name of EKM User>', SECRET='<HSM partition password>'
   FOR CRYPTOGRAPHIC PROVIDER LunaEKMProvider
   ```

   Where CREDENTIAL and IDENTITY can be any user defined unique name.

   **NOTE:** If the partition on a PED based SafeNet PCI HSM not having a partition challenge, then the value for SECRET should be a space character (e.g., ` `), else enter the partition challenge.

2. To map the LunaEKMCredential with SQL User or Login:

   ```
   ALTER LOGIN [Domain\Login Name]
   ADD CREDENTIAL <Name of Credential created>
   ```
NOTE: EKM session needs to be re-opened in case the user changes the HSM slot or the client machine is deleted from SafeNet Luna HSM and registered again or network disconnection.

Using Luna EKM Provider

The Luna EKM provider is now ready to use, it can be used to create/drop symmetric and asymmetric keys to/from the Luna partition and can perform encryption/decryption using these keys. Follow the below steps to exercise the cryptographic capabilities of SafeNet Luna HSM from the SQL Server:

Creating Symmetric Keys on SafeNet Luna HSM

Following types of symmetric key can be created on SafeNet Luna HSM from the SQL Server:

- RC2
- RC4*
- RC4_128*
- DES
- Triple_DES
- Triple_DES_3KEY
- AES_128
- AES_192
- AES_256

* Deprecated in SQL Server 2012.

In the examples below, AES algorithm is used for the symmetric key operation. In order to test other algorithms, AES ALGORITHM tag can be replaced with any of the other tags from the above list.

To create the symmetric key using Luna EKM Provider:

1. Execute the following command from the SQL query window:

   ```
   CREATE SYMMETRIC KEY SQL_EKM_AES_256_Key
   FROM Provider LunaEKMProvider
   WITH ALGORITHM = AES_256,
   PROVIDER_KEY_NAME = 'EKM_AES_256_Key',
   CREATION_DISPOSITION=CREATE_NEW
   ```

   **NOTE:** Once a key is created on the SafeNet Luna HSM, it can be used or referred by its name from the SQL Server, for example in the above said test case, SQL_EKM_AES_256_Key is the unique name of the key in the SQL Server which can be used to perform crypto operation (encrypt/decrypt) using the key on the SafeNet Luna HSM.
Viewing Symmetric Keys

To view the symmetric keys for Luna EKM Provider:

1. Execute the following command from the SQL query window:
   
   ```sql
   SELECT * FROM [master].[sys].[symmetric_keys]
   ```

Encryption using Symmetric Keys

To encrypt using symmetric key:

1. Create a test Table in the MASTER database with fields.

   ```sql
   CREATE TABLE test(
   id numeric(10),
   name varchar (50),
   data varchar (max),)
   ```

2. Execute the following command from the SQL query window:

   ```sql
   INSERT INTO dbo.test
   values( 1,'some text',
   EncryptByKey(Key_GUID('SQL_EKM_AES_256_Key'), 'text to be encrypted'))
   ```

Decryption using Symmetric Keys

To decrypt using symmetric key:

1. Execute the following command from the SQL query window:

   ```sql
   SELECT id,name,CONVERT(varchar(MAX),
   DecryptByKey(data))
   FROM dbo.test where id =1
   ```

Dropping Symmetric Keys

To drop the symmetric key:

1. Execute the following command from the SQL query window:

   ```sql
   DROP SYMMETRIC KEY SQL_EKM_AES_256_Key REMOVE PROVIDER KEY
   ```

   This command drops the key from the SQL Server as well as from the SafeNet Luna HSM.

Creating Asymmetric Keys on SafeNet Luna HSM

Following types of asymmetric key can be created on SafeNet Luna HSM from the SQL Server:

- RSA_512
- RSA_1024
- RSA_2048
- RSA_3072
- RSA_4096

In the examples below, RSA_2048 algorithm is used for the asymmetric key operation. In order to test other algorithms, RSA_2048 ALGORITHM tag can be replaced with any of the other tags from the above list.
To create the asymmetric key using Luna EKM Provider:

1. Execute the following command from the SQL query window:
   
   ```
   CREATE ASYMMETRIC KEY SQL_EKM_RSA_2048_Key
   FROM Provider LunaEKMProvider
   WITH ALGORITHM = RSA_2048,
   PROVIDER_KEY_NAME = 'EKM_RSA_2048_Key',
   CREATION_DISPOSITION=CREATE_NEW
   ```

   **NOTE:** Once a key is created on the SafeNet Luna HSM, it can be used or referred by its name from the SQL Server, for example in the above said test case, SQL_EKM_RSA_2048_Key is the unique name of the key in the SQL Server which can be used to perform crypto operation (encrypt/decrypt) using the key on the SafeNet Luna HSM.

**Viewing Asymmetric Keys**

To view the asymmetric keys for Luna EKM Provider:

1. Execute the following command from the SQL query window:
   
   ```
   SELECT * FROM [master].[sys].[asymmetric_keys]
   ```

**Encryption using Asymmetric Keys**

To encrypt using asymmetric key:

1. Create a test Table in the MASTER database with fields:
   
   ```
   Create Table test(
   id numeric(10),
   name varchar (50),
   data varchar (max),)
   ```

2. Execute the following command from the SQL query window:
   
   ```
   INSERT INTO dbo.test
   values ( 1,'some text',
   EncryptByAsymKey (AsymKey_Id ('SQL_EKM_RSA_2048_Key'), 'text to be encrypted'))
   ```

**Decryption using Asymmetric Keys**

To decrypt using asymmetric key:

1. Execute the following command from the SQL query window:
   
   ```
   SELECT id, name, CONVERT (varchar (MAX),
   DecryptByAsymKey (AsymKey_Id ('SQL_EKM_RSA_2048_Key'), data))
   FROM dbo.test where id =1
   ```

**Dropping Asymmetric Keys**

To drop the asymmetric key:

1. Execute the following command from the SQL query window:
   
   ```
   DROP ASYMMETRIC KEY SQL_EKM_RSA_2048_Key REMOVE PROVIDER KEY
   ```
This command drops the key from the SQL Server as well as from the SafeNet Luna HSM.

Creating Symmetric Key Encrypted by Asymmetric Key on SafeNet Luna HSM

To create a symmetric Key encrypted by an asymmetric Key on HSM

1. Execute the following command from SQL query window:

   Create SYMMETRIC KEY key1
   WITH ALGORITHM = AES_256
   ENCRYPTION BY Asymmetric Key SQL_EKM_RSA_2048_Key;

   SQL_EKM_RSA_2048_Key is an existing asymmetric key on SafeNet Luna HSM. To generate the key, refer to the Creating Asymmetric Keys on SafeNet Luna HSM section.

2. Before using the key, you need to open the key. Execute the following command to open the symmetric key:

   OPEN SYMMETRIC KEY key1 DECRYPTION BY Asymmetric Key SQL_EKM_RSA_2048_Key;

3. Create a test Table in the MASTER database with fields:

   Create Table test(
     id numeric(10),
     name varchar (50),
     data varchar (max),)

4. To encrypt using symmetric key, execute the following command from the SQL query window:

   INSERT INTO dbo.test
   values ( 1,'some text',
   Encryptbykey(KEY_GUID('key1'),'text to be encrypted'))

5. To decrypt using symmetric key, execute the following command from the SQL query window:

   SELECT id,name,CONVERT(varchar(MAX),
   DecryptByKey(data))
   FROM dbo.test where id =1

6. Close the symmetric key by executing the following command:

   CLOSE SYMMETRIC KEY key1

Enable Transparent Database Encryption using Asymmetric key on SafeNet Luna HSM

To enable Transparent Database Encryption using asymmetric key on HSM, execute the following steps:

1. Create an asymmetric key using Luna EKM Provider.

   Use master;
   CREATE ASYMMETRIC KEY SQL_EKM_RSA_2048_Key_TDE
   FROM Provider LunaEKMProvider
   WITH ALGORITHM = RSA_2048,
   PROVIDER_KEY_NAME = 'EKM_RSA_2048_Key_TDE',
   CREATION_DISPOSITION=CREATE_NEW

2. Create a credential for Luna EKM Provider.

   CREATE CREDENTIAL <Name of credential>
   WITH IDENTITY='<Name of EKM User>', SECRET='<HSM partition password>'
   FOR CRYPTOGRAPHIC PROVIDER LunaEKMProvider

3. Create a login based on the asymmetric key created above.

   CREATE LOGIN <Name of login>
FROM ASYMMETRIC KEY SQL_EKM_RSA_2048_Key_TDE;

4. Map the credential created above to the login created above.
   ALTER LOGIN <Name of Login>
   ADD CREDENTIAL <Name of credential>;

5. Create a Database Encryption Key.
   
   **NOTE:** Database encryption operations cannot be performed for 'master', 'model', 'tempdb', 'msdb', or 'resource' databases.

   CREATE DATABASE TDE;
   Use tde;
   CREATE DATABASE ENCRYPTION KEY
   WITH ALGORITHM = AES_256
   ENCRYPTION BY SERVER ASYMMETRIC KEY SQL_EKM_RSA_2048_Key_TDE;

   ALTER DATABASE TDE
   SET ENCRYPTION ON;

7. To query the status of database encryption and its percentage completion.
   SELECT DB_NAME (e.database_id) AS DatabaseName,
       e.database_id,
       e.encryption_state,
       CASE e.encryption_state
           WHEN 0 THEN 'No database encryption key present, no encryption'
           WHEN 1 THEN 'Unencrypted'
           WHEN 2 THEN 'Encryption in progress'
           WHEN 3 THEN 'Encrypted'
           WHEN 4 THEN 'Key change in progress'
           WHEN 5 THEN 'Decryption in progress'
       END AS encryption_state_desc,
       c.name,
       e.percent_complete
   FROM sys.dm_database_encryption_keys AS e
   LEFT JOIN master.sys.asymmetric_keys AS c
   ON e.encryptor_thumbprint = c.thumbprint

**Transparent Database Encryption (TDE) Key Rotation**

1. Create an asymmetric key using Luna EKM Provider.
   Use master;
   CREATE ASYMMETRIC KEY SQL_EKM_RSA_2048_Key_TDE_Rot
   FROM Provider LunaEKMProvider
   WITH ALGORITHM = RSA_2048,
   PROVIDER_KEY_NAME = 'EKM_RSA_2048_Key_TDE_Rot',
   CREATION_DISPOSITION=CREATE_NEW

2. Create a credential for Luna EKM Provider.
   CREATE CREDENTIAL <Name of credential>
   WITH IDENTITY='<Name of EKM User>', SECRET='<HSM partition password>'
   FOR CRYPTOGRAPHIC PROVIDER LunaEKMProvider
3. Create a login based on the asymmetric key created above.
   
   CREATE LOGIN <Name of login>
   FROM ASYMMETRIC KEY SQL_EKM_RSA_2048_Key_TDE_Rot;

4. Map the credential created above to the login created above.
   
   ALTER LOGIN <Name of Login>
   ADD CREDENTIAL <Name of credential>;

5. Enable Transparent Database Encryption Key Rotation.
   
   Use tde;
   ALTER DATABASE ENCRYPTION KEY
   REGENERATE
   WITH ALGORITHM = AES_128
   ALTER DATABASE ENCRYPTION KEY
   ENCRYPTION BY SERVER ASYMMETRIC KEY SQL_EKM_RSA_2048_Key_TDE_Rot;
   
   go
   SELECT * FROM sys.dm_database_encryption_keys
   go

6. To query the status of database encryption, TDE key change and its percentage completion.
   
   SELECT DB_NAME (e.database_id) AS DatabaseName,
   e.database_id,
   e.encryption_state,
   CASE e.encryption_state
   WHEN 0 THEN 'No database encryption key present, no encryption'
   WHEN 1 THEN 'Unencrypted'
   WHEN 2 THEN 'Encryption in progress'
   WHEN 3 THEN 'Encrypted'
   WHEN 4 THEN 'Key change in progress'
   WHEN 5 THEN 'Decryption in progress'
   END AS encryption_state_desc,
   c.name,
   e.percent_complete
   FROM sys.dm_database_encryption_keys AS e
   LEFT JOIN master.sys.asymmetric_keys AS c
   ON e.encryptor_thumbprint = c.thumbprint

---

**Migration from SQL EKM to Luna EKM**

If you have enabled the Transparent Data Encryption for any database (let AdventureWorks) using the SQL EKM and now want to migrate TDE from SQL EKM to Luna EKM. Previously your database master key is encrypted by either Certificate or Asymmetric Key which was generated in SQL and after enabling the TDE with Luna EKM. Now you want to generate a new database master key encrypted by asymmetric key generated on Luna HSM. To do this you need to perform the following steps:

**NOTE:** We have tested these steps in SQL Server 2014 with Luna Client 5.4.1.

---

1. Decrypt the database (let AdventureWorks).
   
   USE master;
   ALTER DATABASE AdventureWorks
2. Take the backup of database and transaction logs. When the backup completed restart the SQL database.

3. Create an asymmetric key using Luna EKM Provider.

   Use master;
   CREATE ASYMMETRIC KEY SQL_EKM_RSA_2048_Key_AW
   FROM Provider LunaEKMProvider
   WITH ALGORITHM = RSA_2048,
   PROVIDER_KEY_NAME = 'EKM_RSA_2048_Key_AW',
   CREATION_DISPOSITION=CREATE_NEW

4. Create a credential for Luna EKM Provider.

   CREATE CREDENTIAL <Name of credential>
   WITH IDENTITY='<Name of EKM User>', SECRET='<HSM partition password>'
   FOR CRYPTOGRAPHIC PROVIDER LunaEKMProvider

5. Create a login based on the asymmetric key created above.

   CREATE LOGIN <Name of login>
   FROM ASYMMETRIC KEY SQL_EKM_RSA_2048_Key_AW;

6. Map the credential created above to the login created above.

   ALTER LOGIN <Name of Login>
   ADD CREDENTIAL <Name of credential>;

7. Create or Regenerate a Database Encryption Key.

   USE AdventureWorks;
   CREATE DATABASE ENCRYPTION KEY
   WITH ALGORITHM = AES_256
   ENCRYPTION BY SERVER ASYMMETRIC KEY SQL_EKM_RSA_2048_Key_AW;
   Or

   USE AdventureWorks;
   ALTER DATABASE ENCRYPTION KEY
   REGENERATE WITH ALGORITHM = AES_192
   ENCRYPTION BY SERVER ASYMMETRIC KEY SQL_EKM_RSA_2048_Key_AW;

8. Enable Transparent Database Encryption.

   ALTER DATABASE AdventureWorks
   SET ENCRYPTION ON;

9. To query the status of database encryption and its percentage completion.

   SELECT DB_NAME(e.database_id) AS DatabaseName,
   e.database_id,
   e.encryption_state,
   CASE e.encryption_state
   WHEN 0 THEN 'No database encryption key present, no encryption'
   WHEN 1 THEN 'Unencrypted'
   WHEN 2 THEN 'Encryption in progress'
   WHEN 3 THEN 'Encrypted'
   WHEN 4 THEN 'Key change in progress'
   WHEN 5 THEN 'Decryption in progress'
   END AS encryption_state_desc,
   c.name,
   e.percent_complete
Using Extensible Key Management on a SQL Server Failover Cluster

This section focuses on the preparation of the environment for 2-node SQL Server Cluster in Windows Server 2008 R2.

1. Refer to the SQL Server documentation to install a failover cluster.

   **Setting up a Shared Storage**

   To set up a shared storage disk for SQL Server Cluster, refer to the configuration procedures that apply for shared storage solution. Plan the size of the shared storage depending on the number of certificates that are required to be enrolled.

2. Once the cluster is up and running, install SafeNet Luna Network HSM client on both the nodes.

3. Configure and setup the appliance on both the nodes and register the same partition on both node of SQL Server Cluster.

4. Install Luna EKM client on both the nodes.

5. Configure the Luna EKM provider on both the nodes.
6. Open the SQL Server management studio to register the Luna EKM provider on the first node.
7. Setup the credential on the first node.
8. Create some keys using the Luna EKM provider on the first node.
9. Create a table and encrypt some column with the Luna EKM key with the first node.
10. Shutdown the first node.
11. Login to the second node and decrypt the data encrypted on the first node.
12. Data is decrypted successfully.

Extensible Key Management using Luna EKM is working fine on a SQL Server cluster.
The Always On Availability Groups feature is a high-availability and disaster recovery solution that provides an enterprise-level alternative to database mirroring. Introduced in SQL Server 2012, Always On Availability Groups maximizes the availability of a set of user databases for an enterprise. An availability group supports a failover environment for a discrete set of user databases, known as availability databases that fail over together. An availability group supports a set of read-write primary databases and one to eight sets of corresponding secondary databases. Optionally, secondary databases can be made available for read-only access and/or some backup operations.

To perform SafeNet Luna HSM integration with SQL Server, Luna EKM software provides Luna EKM Provider in the form of EKM Library (i.e. LunaEKM.dll). The Luna EKM Provider can be used if the EKM Provider option is enabled in the SQL Server. This feature is available only on the Enterprise, Developer, and Evaluation editions of SQL Server. By default, Extensible Key Management is off.

**NOTE:** To setup Luna Client and Luna EKM, refer to Chapter 1. Luna Client and Luna EKM must be setup on all nodes of SQL Server which needs to be added in the “Always On” availability group. All nodes must be registered with the same partition of SafeNet Luna HSM.

**SQL Server Setup**

SQL Server must be installed on the target machines to carry out the integration process and all nodes have WFCS. For a detailed installation procedure of SQL Server, Always On group pre-requisites, and recommendations refer to the Microsoft SQL Server online documentation.

**Enabling EKM Provider Option**

To enable this feature, use the `sp_configure` command on both the SQL Server nodes that has the following option and value, as in the following example:

To enable the Extensible Key Management option:

1. Open the SQL Server Management Studio.
2. Connect to the SQL Server.
3. Open a query window, and then run the following command:
   ```sql
   sp_configure 'show advanced', 1
   GO
   RECONFIGURE
   GO
   ```
Registering Luna EKM Provider

To setup the Luna EKM provider, Luna EKM Software must be installed and needs to be registered with the SQL Server. Follow the below steps to create/register the provider on all SQL Server nodes:

To create/register the Luna EKM Provider:

1. Open the SQL Server Management Studio.
2. Connect to the SQL Server.
3. Open a query window, and then run the following command:
   ```sql
   CREATE CRYPTOGRAPHIC PROVIDER <Name of Cryptographic Provider>
   FROM FILE = '<Location of Luna EKM Provider Library>'
   ``

   where CRYPTOGRAPHIC PROVIDER can be any user defined unique name.

4. To view the list of EKM providers:
   ```sql
   SELECT [provider_id], [name], [guid], [version], [dll_path], [is_enabled]
   FROM [model].[sys].[cryptographic_providers]
   ``

5. To view the provider properties:
   ```sql
   SELECT [provider_id], [guid], [provider_version], [sqlcrypt_version], [friendly_name], [authentication_type], [symmetric_key_support], [symmetric_key_persistance], [symmetric_key_export], [symmetric_key_import], [asymmetric_key_support], [asymmetric_key_persistance], [asymmetric_key_export], [asymmetric_key_import]
   FROM [master].[sys].[dm_cryptographic_provider_properties]
   ```

Setting up Credential for Luna EKM Provider

The next step is to create a CREDENTIAL for the Luna EKM Provider. The CREDENTIAL must be mapped to SQL User or login to be able to use the Luna EKM Provider. A CREDENTIAL is basically used to access any
external SQL Server resource such as SafeNet Luna HSM. Follow the below steps to create\map credential for the provider on all SQL Server nodes:

To create\map the CREDENTIAL for Luna EKM Provider:

1. Open a query window, and run the following command:

   CREATE CREDENTIAL <Name of credential>
   WITH IDENTITY='<Name of EKM User>', SECRET='<HSM partition password>'
   FOR CRYPTOGRAPHIC PROVIDER LunaEKMProvider

   Where CREDENTIAL and IDENTITY can be any user defined unique name.

2. To map the LunaEKMCred with SQL User or Login:

   ALTER LOGIN [Domain\Login Name]
   ADD CREDENTIAL <Name of Credential created>

---

**NOTE:** It is assumed that a common user is used on all SQL Server nodes that become the part of “Always On” availability group. Above steps (listed under Enabling EKM Provider option, Registering Luna EKM Provider, and Setting up Credential for Luna EKM Provider) needs to be executed on all secondary nodes as well.

---

**NOTE:** EKM session needs to be re-opened in case where user changes the SafeNet Luna HSM slot or the client machine is deleted from SafeNet Luna HSM and registered again or network disconnection.

---

**Creating the Always On Availability Group**

Open the Microsoft SQL Server Management Studio on primary replica and create a database and then, take full backup of that database on a shared location that is accessible by all SQL Server nodes. Open the Always On Availability Group Creation wizard and follow the instructions to create the Always On Availability Group. For detailed steps and prerequisites, refer the Microsoft online documentation for creating the Always On Availability Group.
After the successful creation of the group, the dashboard displays all the participating nodes. An example of a dashboard is shown below. For demo purpose, two nodes are added; primary and secondary.

Creating the Encryption Keys for Availability Group Database

Creating Symmetric Keys on SafeNet Luna HSM

Following types of symmetric key can be created on SafeNet Luna HSM from the SQL Server:

- RC2
- RC4*
- RC4_128*
- DES
- Triple_DES
- Triple_DES_3KEY
- AES_128
- AES_192
- AES_256

* Deprecated in SQL Server 2012.
In the examples below, AES algorithm is used for the symmetric key operation. In order to test other algorithms, AES ALGORITHM tag can be replaced with any of the other tags from the above list.

To create the symmetric key using Luna EKM Provider on availability database (e.g. HSMDB) open the SMS on primary replica:

1. Execute the following command from the SQL query window:
   USE HSMDB;
2. Execute the following command from the SQL query window:
   CREATE SYMMETRIC KEY SQL_EKM_AES_256_Key
   FROM Provider LunaEKMP\r\n   WITH ALGORITHM = AES_256,
   PROVIDER_KEY_NAME = 'EKM_AES_256_Key',
   CREATION_DISPOSITION=CREATE_NEW

   **NOTE:** Once a key is created on the SafeNet Luna HSM, it can be used or referred by its name from the SQL Server, for example in the above said test case, SQL_EKM_AES_256_Key is the unique name of the key in the SQL Server that is used to perform crypto operation (encrypt/decrypt) using the key on the SafeNet Luna HSM.

**Viewing Symmetric Keys**

To view the symmetric keys for Luna EKM Provider:

1. Execute the following command from the SQL query window:
   SELECT * FROM [hsmdb].[sys].[symmetric_keys]

**Encryption using Symmetric Keys**

To encrypt using symmetric key:

1. Create a test Table in the HSMDB database with fields.
   Create Table test(
   id numeric(10),
   name varchar (50),
   data varchar (max),)
2. Execute the following command from the SQL query window:
   INSERT INTO dbo.test
   values( 1,'some text',
   EncryptByKey(Key_GUID('SQL_EKM_AES_256_Key'), 'text to be encrypted'))

**Decryption using Symmetric Keys**

To decrypt using symmetric key:

1. Execute the following command from the SQL query window:
   SELECT id,name,CONVERT(varchar(MAX),
   DecryptByKey(data))
   FROM dbo.test where id =1
2. Now execute the above command on secondary replica and verify that the output is same as primary replica.

**Creating Asymmetric Keys on SafeNet Luna HSM**

Following types of asymmetric key can be created on SafeNet Luna HSM from the SQL Server:

- RSA_512
- RSA_1024
- RSA_2048

In the examples below, RSA_2048 algorithm is used for the asymmetric key operation. In order to test other algorithms, RSA_2048 ALGORITHM tag can be replaced with any of the other tags from the above list.

To create the asymmetric key using Luna EKM Provider open the SMS on primary replica:

1. Execute the following command from the SQL query window:

```
CREATE ASYMMETRIC KEY SQL_EKM_RSA_2048_Key
FROM Provider LunaEKMProvider
WITH ALGORITHM = RSA_2048,
PROVIDER_KEY_NAME = 'EKM_RSA_2048_Key',
CREATION_DISPOSITION=CREATE_NEW
```

**NOTE:** Once a key is created on the SafeNet Luna HSM, it can be used or referred by its name from the SQL Server, for example in the above said test case, SQL_EKM_RSA_2048_Key is the unique name of the key in the SQL Server which can be used to perform crypto operation (encrypt/decrypt) using the key on the SafeNet Luna HSM.

**Viewing Asymmetric Keys**

To view the asymmetric keys for Luna EKM Provider:

1. Execute the following command from the SQL query window:

```
SELECT * FROM [hsmdb].[sys].[asymmetric_keys]
```

**Encryption using Asymmetric Keys**

To encrypt using asymmetric key:

1. Execute the following command from the SQL query window:

```
INSERT INTO dbo.test
values ( 2,'some text',
EncryptByAsymKey (AsymKey_Id ('SQL_EKM_RSA_2048_Key'), 'text to be encrypted'))
```

**Decryption using Asymmetric Keys**

To decrypt using asymmetric key:

1. Execute the following command from the SQL query window:

```
SELECT id, name, CONVERT (varchar (MAX),
DecryptByAsymKey (AsymKey_Id ('SQL_EKM_RSA_2048_Key'), data))
```
FROM dbo.test where id =2

2. Now execute the above command on secondary replica and verify that the output is same as primary replica.

Creating Symmetric Key Encrypted by Asymmetric Key on SafeNet Luna HSM

To create a symmetric key encrypted by an asymmetric key on SafeNet Luna HSM, open the SMS on primary replica:

1. Execute the following command from SQL query window:
   
   Create SYMMETRIC KEY key1
   WITH ALGORITHM = AES_256
   ENCRYPTION BY Asymmetric Key SQL_EKM_RSA_2048_Key;

   where “SQL_EKM_RSA_2048_Key” is an existing asymmetric key.

2. Before using the key you need to open the key. Following command can be executed to open the symmetric key.
   
   OPEN SYMMETRIC KEY key1 DECRYPTION BY Asymmetric Key SQL_EKM_RSA_1024_Key;

3. Encrypt the data using the key1:
   
   INSERT INTO dbo.test
   values ( 3,'some text',
   EncryptByKey(KEY_GUID('Key1'), 'text to be encrypted'))

4. Decrypt the data using the key1:
   
   SELECT id,name,CONVERT(varchar(MAX),
   DecryptByKey(data))
   FROM dbo.test where id =3

5. Close the symmetric key by executing the following command:
   
   CLOSE SYMMETRIC KEY key1

6. Now execute the above steps (2-5) on secondary replica and verify that the output is same as primary replica.

Enable Transparent Database Encryption using Asymmetric key on SafeNet Luna HSM

To enable Transparent Database Encryption using asymmetric key on SafeNet Luna HSM, open the SMS on primary replica:

1. Create an asymmetric key using Luna EKM Provider on primary replica.
   
   Use master;
   CREATE ASYMMETRIC KEY SQL_EKM_RSA_2048_Key_TDE
   FROM Provider LunaEKMProvider
   WITH ALGORITHM = RSA_2048,
   PROVIDER_KEY_NAME = 'EKM_RSA_2048_Key_TDE',
   CREATION_DISPOSITION=CREATE_NEW

2. Create same asymmetric key using Luna EKM Provider on secondary replica.
   
   Use master;
   CREATE ASYMMETRIC KEY SQL_EKM_RSA_2048_Key_TDE
   FROM Provider LunaEKMProvider
   WITH PROVIDER_KEY_NAME = 'EKM_RSA_2048_Key_TDE',
   CREATION_DISPOSITION=OPEN_EXISTING
3. Create a credential for Luna EKM Provider.
   
   ```sql
   CREATE CREDENTIAL <Name of credential>
   WITH IDENTITY='<Name of EKM User>', SECRET='<HSM partition password>'
   FOR CRYPTOGRAPHIC PROVIDER LunaEKMProvider
   ```

4. Create a login based on the asymmetric key created above.
   
   ```sql
   CREATE LOGIN <Name of login>
   FROM ASYMMETRIC KEY SQL_EKM_RSA_2048_Key_TDE;
   ```

5. Map the credential created above to the login created above.
   
   ```sql
   ALTER LOGIN <Name of Login>
   ADD CREDENTIAL <Name of credential>;
   ```

6. Execute the above steps (2-5) for all secondary nodes. It is required because TDE encryption key, credential, and login are created in the master database and it is not a part of Availability Groups. Therefore, you need to create the same key, credential and login in the master database of all the secondary nodes to access the encrypted tables created in the database.

7. Create a Database Encryption Key on primary replica.

   **NOTE:** Database encryption operations cannot be performed for 'master', 'model', 'tempdb', 'msdb', or 'resource' databases.

   ```sql
   CREATE DATABASE TDE;
   USE tde;
   CREATE DATABASE ENCRYPTION KEY
   WITH ALGORITHM = AES_256
   ENCRYPTION BY SERVER ASYMMETRIC KEY SQL_EKM_RSA_2048_Key_TDE;
   ```

8. Enable Transparent Database Encryption:
   
   ```sql
   ALTER DATABASE TDE
   SET ENCRYPTION ON;
   ```

9. To query the status of database encryption and its percentage completion.
   
   ```sql
   SELECT DB_NAME (e.database_id) AS DatabaseName,
          e.database_id,
          e.encryption_state,
          CASE e.encryption_state
              WHEN 0 THEN 'No database encryption key present, no encryption'
              WHEN 1 THEN 'Unencrypted'
              WHEN 2 THEN 'Encryption in progress'
              WHEN 3 THEN 'Encrypted'
              WHEN 4 THEN 'Key change in progress'
              WHEN 5 THEN 'Decryption in progress'
              END AS encryption_state_desc,
          c.name,
          e.percent_complete
   FROM sys.dm_database_encryption_keys AS e
   LEFT JOIN master.sys.asymmetric_keys AS c
   ON e.encryptor_thumbprint = c.thumbprint
   ```
Add the encrypted database in to the availability group

Before adding the already encrypted database into availability group, take the full backup of the database on shared location that is accessible by all secondary nodes.

To add the encrypted database into the availability group, open the SMS on primary replica:

1. Add the database (e.g. TDE) into the availability group (e.g. AGroup).
   
   ```
   use master;
   ALTER AVAILABILITY GROUP AGroup ADD DATABASE tde;
   GO
   
   Above command add the database into the availability group but it is not available on secondary replica. To synchronize the database you need to restore the database on secondary replica.
   ```

2. Restore the database on secondary replica from the location where you have backed up with "RESTORE WITH NORECOVERY" option.

3. Add the database on secondary replica using the following SQL command:
   
   ```
   use master;
   ALTER DATABASE tde SET HADR AVAILABILITY GROUP = AGroup;
   ```

4. To query the status of database encryption and its percentage completion on secondary node.
   
   ```
   SELECT DB_NAME (e.database_id) AS DatabaseName, 
   e.database_id, 
   e.encryption_state, 
   CASE e.encryption_state 
   WHEN 0 THEN 'No database encryption key present, no encryption'
   WHEN 1 THEN 'Unencrypted' 
   WHEN 2 THEN 'Encryption in progress'
   WHEN 3 THEN 'Encrypted'
   WHEN 4 THEN 'Key change in progress'
   WHEN 5 THEN 'Decryption in progress'
   END AS encryption_state_desc, 
   c.name, 
   e.percent_complete 
   FROM sys.dm_database_encryption_keys AS e 
   LEFT JOIN master.sys.asymmetric_keys AS c 
   ON e.encryptor_thumbprint = c.thumbprint
   ```

Transparent Database Encryption (TDE) Key Rotation

1. Create an asymmetric key using Luna EKM Provider on primary replica.
   
   ```
   Use master;
   CREATE ASYMMETRIC KEY SQL_EKM_RSA_2048_Key_TDE_Rot
   FROM Provider LunaEKMProvider 
   WITH ALGORITHM = RSA_2048, 
   PROVIDER_KEY_NAME = 'EKM_RSA_2048_Key_TDE_Rot', 
   CREATION_DISPOSITION=CREATE_NEW
   ```

2. Create same asymmetric key using Luna EKM Provider on secondary replica.

   ```
   Use master;
   CREATE ASYMMETRIC KEY SQL_EKM_RSA_2048_Key_TDE_Rot
   FROM Provider LunaEKMProvider 
   WITH PROVIDER_KEY_NAME = 'EKM_RSA_2048_Key_TDE_Rot', 
   CREATION_DISPOSITION=OPEN_EXISTING
3. Create a credential for Luna EKM Provider.

   CREATE CREDENTIAL <Name of credential>
   WITH IDENTITY='<Name of EKM User>', SECRET='<HSM partition password>'
   FOR CRYPTOGRAPHIC PROVIDER LunaEKMProvider

4. Create a login based on the asymmetric key created above.

   CREATE LOGIN <Name of login>
   FROM ASYMMETRIC KEY SQL_EKM_RSA_2048_Key_TDE_Rot;

5. Map the credential created above to the login created above.

   ALTER LOGIN <Name of Login>
   ADD CREDENTIAL <Name of credential>;

6. Execute steps 2-5 for all secondary nodes. It is required because TDE encryption key, credential, and login are created in the master database and it is not a part of Availability Groups. Therefore, you need to create the same key, credential, and login in the master database of all the secondary nodes to access the encrypted tables created in the database.

7. Enable Transparent Database Encryption Key Rotation on the primary replica.

   Use tde;
   ALTER DATABASE ENCRYPTION KEY
   REGENERATE
   WITH ALGORITHM = AES_128

   ALTER DATABASE ENCRYPTION KEY
   ENCRYPTION BY SERVER ASYMMETRIC KEY SQL_EKM_RSA_2048_Key_TDE_Rot;
   go
   SELECT * FROM sys.dm_database_encryption_keys
   go

8. To query the status of database encryption, TDE key change and its percentage completion.

   SELECT DB_NAME (e.database_id) AS DatabaseName,
   e.database_id,
   e.encryption_state,
   CASE e.encryption_state
   WHEN 0 THEN 'No database encryption key present, no encryption'
   WHEN 1 THEN 'Unencrypted'
   WHEN 2 THEN 'Encryption in progress'
   WHEN 3 THEN 'Encrypted'
   WHEN 4 THEN 'Key change in progress'
   WHEN 5 THEN 'Decryption in progress'
   END AS encryption_state_desc,
   c.name,
   e.percent_complete
   FROM sys.dm_database_encryption_keys AS e
   LEFT JOIN master.sys.asymmetric_keys AS c
   ON e.encryptor_thumbprint = c.thumbprint
Overview

The new feature, called Always Encrypted, is available from SQL Server 2016’s first public preview. Always Encrypted adds an extra measure of security when the data is being used. It is the point at which data can be most susceptible to attack. The new security layer addresses that vulnerability by keeping the data encrypted during transactions and computations, and by only giving the client keys to decrypt it. It means that if anyone else, including a database or system administrator, tries to access that client’s database, the credit card information or other sensitive data would appear as a gibberish value. In order to use SQL Always Encrypted, the following keys are created:

- Column master key
- Column encryption key

A column encryption key is used to encrypt data in an encrypted column. A column master key is a key-protecting key that encrypts one or more column encryption keys.

The Database Engine stores encryption configuration for each column in database metadata. However, the Database Engine never stores or uses the keys of either type in plaintext. It only stores encrypted values of column encryption keys and the information about the location of column master keys, which are stored in external trusted key stores, such as Hardware Security Module (HSM).

The feature is depicted in the figure below:
Setting up SQL Server Always Encrypted using SafeNet Luna HSM

This guide demonstrates the method to create the column master key on a SafeNet Luna HSM and encrypt column encryption keys.

Configure SafeNet KSP

To generate the column master key and Encryption Key on SafeNet Luna HSM, SafeNet Luna KSP must be registered on the target machine. Use the following steps to register Luna KSP:

1. Browse to the directory where **32-bit Luna KSP** is installed. The general form of the path is:
   
   `<Luna Client Installation Directory>\win32\KSP`
   
   **For Example:** cd "C:\Program Files\SafeNet\LunaClient\win32\KSP"

2. Run KSPConfig.exe (KSP configuration wizard) utility to register Luna KSP through GUI. The general form of command is:
   
   `<Luna Client Installation Directory>\win32\KSP> KspConfig.exe`
   
   **For Example:** \C:\Program Files\SafeNet\LunaClient\win32\KSP>KspConfig.exe

3. Double-click **Register or View Security Library** on the left side of the pane.

4. Browse the library `<Luna Client installation Directory>\win32\cryptoki.dll or cKlog.dll library` and click **Register**.

5. On successful registration, a message **"Success registering the security library"** displays.

6. Double-click **Register HSM Slots** on the left side of the pane.

7. Verify in **Available Slots** drop down, the partition is listed. Enter the partition password in **Slot Password** field.

8. Click **Register Slot** to register the slot for Domain\User. On successful registration, a message **"The slot was successfully and securely registered"** displays.

9. You need to register the slot for **NT_AUTHORITY\SYSTEM**. Select **NT_AUTHORITY** from drop down under **Register For User** and select **SYSTEM** from drop down under **Domain**.

10. Verify in **Available Slots** drop down, the partition is listed. Enter the partition password in **Slot Password** field.

11. Click **Register Slot** to register the slot for **NT_AUTHORITY\SYSTEM**. On successful registration, a message **"The slot was successfully and securely registered"** displays.

Generating Column Master Key

12. On Client machine, connect to the database using **SQL Server Management Studio**.

13. In Object Explorer, navigate to **Databases > Test > Security**.

---

**NOTE:** Test is the sample database created for demonstration purpose.
14. Expand the **Always Encrypted Keys** folder to display its two subfolders:

- **Column Master Keys**
- **Column Encryption Keys**

15. Right Click on **Column Master Keys** and select **New Column Master Key** to open the following wizard:

![New Column Master Key Wizard](image)

16. Specify the name of the column master key pair to be generated in **Name** field.

17. Select **Key Store** as **Key Storage Provider (CNG)**.

18. From the drop down menu, select provider "**SafeNet Key Storage Provider**".
19. Click **Generate Key**.
20. The key pair (column master key) is generated on the Luna HSM partition.

![Image of Luna HSM interface](image)

**Generating Column Encryption Key**

21. In Object Explorer, navigate to **Databases > Test > Security**.

22. Expand the **Always Encrypted Keys** folder to display its two subfolders:
   - **Column Master Keys**
   - **Column Encryption Keys**
23. Right Click on Column Encryption Keys and select New Column Encryption Key to open the following wizard:

![New Column Encryption Key Wizard](image)

24. Specify the name of the Column Encryption Key to be generated in the Name field.
25. From drop down, select name of **Column master key**. In this case, select *LUNAKEY*.

![New Column Encryption Key](image)

26. Click **OK**.
27. The key is generated and stored in SQL Server instance where Always Encrypted is implemented.

![SQL Server Management Studio](image.png)

**Implementing Always Encrypted**

1. Create a table `Employee` with fields in database `Test`.
   ```sql
   use Test;
   Create Table Employee(
     id numeric(10),
     name varchar (50),
     data varchar (max),);
   ```

2. Execute the following command from the SQL query window to insert the values:
   ```sql
   INSERT INTO dbo.Employee
   values( 101,'Emp1','ConfidentialData'),(102,'Emp2','PrivateData');
   ```

3. View the table contents in plain text.
   ```sql
   Select * from dbo.Employee;
   ```

4. Right click the table `Employee` to encrypt details and select Encrypt Columns.
5. The **Always Encrypted** wizard displays.

6. The introduction page displays. Click **Next** to proceed.
7. The **Column Selection** page allows to select columns to encrypt and encryption type: Randomized or Deterministic.

For table Employee, choose **Randomized** for *data* column and **Deterministic** for *id* column.

8. Click **Next**. In the **Master Key Configuration** page, confirm message "No additional configuration is necessary because you are using existing keys." Click **Next**.

9. Select **Proceed to finish now** in the **Run Settings** page and click **Next**.

10. Verify the details on **Summary** page and click **Finish** to complete the encryption process.
11. Once the process is completed, click **Close**.

![Screenshot of Always Encrypted wizard log report]

12. Now the selected columns of table Employee (id, data) appear in encrypted form.

   ```sql
   Select * from dbo.Employee;
   ```

**Viewing Data with Always Encrypted Implemented in SQL SSMS**

1. Select the **Query** Menu tab in SSMS, point to **Connection**.
2. Click **Change Connection**. This launches the **Connect to Database Engine** dialog box.
3. Click **Options** and then go to the **Additional Connection Parameters** tab, as shown below:
4. On this tab, type the following value:
   Column Encryption Setting=Enabled

5. After you type the setting, click **Connect**, and then rerun the following query:
   ```
   use Test;
   Select * from dbo.Employee;
   ```

6. Pop-up screen prompts for **Parameterization for Always Encrypted**. Choose your option and proceed.

   Now unencrypted data of table displays.

You have configured SQL Server Always Encrypted feature while securing column master key in SafeNet Luna HSM and encrypting column encryption key with the master key.
Troubleshooting Tips

Problem – 1
Failed to verify Authenticode signature on DLL "C:\Program Files\LunaPCI\EKM\LunaEKM.dll".

Solution
This error could appear in SQL logs if the certificate in the signature of dll cannot be verified because there are no corresponding certificates for this issuer and therefore it is not trusted.

Go to http://www.verisign.com/support/roots.html and download the all root certificates. Install the certificate and install/import it to Trusted Root Certification Authorities store.

Problem – 2
"The decryption key is incorrect" error when you open a symmetric key that is encrypted by an asymmetric key in SQL Server 2008, SQL Server 2012 or SQL Server 2008 R2 on a computer that is running Windows 8 or Windows Server 2012.

Solution
Download the cumulative update package and apply for SQL Server provided by Microsoft to resolve the issue:

- For SQL Server 2008 SP3 on Windows Server 2012 platform:
  http://support.microsoft.com/kb/2863205
- For SQL Server 2008 R2 SP2 on Windows Server 2012 platform:
  http://support.microsoft.com/kb/2871401
- For SQL Server 2012 on Windows Server 2012 platform:
  http://support.microsoft.com/kb/2867319

Problem – 3
CREATE CRYPTOGRAPHIC PROVIDER EKMProvider FROM FILE = <Path to EKM DLL>' fails with below error on Windows 2012:

Error:
Msg 33029, Level 16, State 1, Line 3
Cannot initialize cryptographic provider. Provider error code: 1. (Failure - Consult EKM Provider for details)
Solution
Reboot the OS server and try to create cryptographic provider. It resolves the above problem.

Problem – 4

When the key is generated on secondary node using \texttt{CREATION\_DISPOSITION=OPEN\_EXISTING}:

\begin{verbatim}
CREATE ASYMMETRIC KEY <Name of the SQL Key>
FROM Provider LunaEKMPprovider
WITH PROVIDER_KEY_NAME = '\<Name of the HSM Key\\>',
CREATION\_DISPOSITION=OPEN\_EXISTING
\end{verbatim}

Thumbprint of the key generated on secondary node of SQL servers (For backup/restore of an encrypted database) is different from original thumbprint of key generated on primary node which gives error while restoring the database on secondary nodes.

Solution

Use the EKM v1.2 and regenerate the key on secondary nodes using the same command. This issue is fixed in EKM v1.2.