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.
CONTENTS

PREFACE ................................................................. 6
Scope ........................................................................... 6
Document Conventions .................................................. 6
Command Syntax and Typeface Conventions ....................... 7
Support Contacts .......................................................... 8
Customer Support Portal ................................................ 8
Telephone Support ...................................................... 8
Email Support ................................................................ 8

CHAPTER 1: Introduction ................................................ 9
Third Party Application Details .......................................... 9
Supported Platforms .................................................... 11
Prerequisites .................................................................. 12
Configure the SafeNet Luna HSM ...................................... 12
Provision your HSM on Demand Service ......................... 13
Constraints on HSMoD Services ........................................ 13
Set up Luna EKM .......................................................... 14
Set up SQL Server ........................................................ 14

CHAPTER 2: Integrating SafeNet HSM with SQL Server ........ 15
Enabling the EKM Provider option .................................... 15
Creating and Registering the Luna EKM Provider .................. 16
Setting up the CREDENTIAL for Luna EKM Provider ............ 16
Using the Luna EKM Provider Option ............................... 17
Creating Symmetric Keys on SafeNet HSM ......................... 17
Creating Asymmetric Keys on SafeNet HSM ....................... 18
Creating Symmetric Key Encrypted by Asymmetric Key on SafeNet HSM ........................................ 19
Enabling Transparent Database Encryption using Asymmetric key on SafeNet HSM ......................... 20
Rotating Keys for Transparent Database Encryption .............. 21
Migrating TDE from SQL EKM to Luna EKM ..................... 23
Using Extensible Key Management on a SQL Server Failover Cluster ........................................ 25

CHAPTER 3: Integrating SafeNet HSM with SQL Server High Availability Group .... 27
Enabling the EKM Provider Option .................................... 27
Creating and Registering the Luna EKM Provider .................. 28
Setting up the CREDENTIAL for Luna EKM Provider ............ 28
Creating the Always On Availability Group .......................... 29
Creating the Encryption Keys for Availability Group Database ........................................ 30
Creating Symmetric Keys on SafeNet HSM ......................... 30
Creating Asymmetric Keys on SafeNet HSM ....................... 31
Creating Symmetric Key Encrypted by Asymmetric Key on SafeNet HSM ........................................ 32
Enabling Transparent Database Encryption using Asymmetric key on SafeNet HSM ................. 33
Adding the Encrypted Database to the Availability Group ........ 35
Rotating Keys for Transparent Database Encryption .......................................................... 35

CHAPTER 4: Integrating SafeNet HSM with SQL Server Always Encrypted .............. 38
    Configuring the SafeNet KSP .................................................................................. 38
    Generating the Column Master Key .................................................................... 39
    Generating the Column Encryption Key ............................................................. 42
    Implementing Always Encrypted ........................................................................ 44
    Viewing Always Encrypted Data ........................................................................... 47

CHAPTER 5: Troubleshooting Tips ............................................................................. 49
PREFACE

This document is intended to guide security administrators through the steps to install, configure and integrate Microsoft SQL Server with a SafeNet Luna HSM or HSM on Demand (HSMoD) service.

Scope

This document outlines the steps to integrate a Microsoft SQL server with a SafeNet Luna HSM or HSM on Demand (HSMoD) service to secure the SQL Server master encryption keys.

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. Notes contain 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>
<tbody>
<tr>
<td><strong>bold</strong></td>
<td>The bold attribute is used to indicate the following:</td>
</tr>
<tr>
<td></td>
<td>&gt; Command-line commands and options (Type <code>dir /p</code>.)</td>
</tr>
<tr>
<td></td>
<td>&gt; Button names (Click <strong>Save As</strong>.)</td>
</tr>
<tr>
<td></td>
<td>&gt; Check box and radio button names (Select the <strong>Print Duplex</strong> check box.)</td>
</tr>
<tr>
<td></td>
<td>&gt; Window titles (On the <strong>Protect Document</strong> window, click <strong>Yes</strong>.)</td>
</tr>
<tr>
<td></td>
<td>&gt; Field names (<strong>User Name</strong>: Enter the name of the user.)</td>
</tr>
<tr>
<td></td>
<td>&gt; Menu names (On the <strong>File</strong> menu, click <strong>Save</strong>.) (Click <strong>Menu &gt; Go To &gt; Folders</strong>.)</td>
</tr>
<tr>
<td></td>
<td>&gt; User input (In the <strong>Date</strong> box, type <strong>April 1</strong>.)</td>
</tr>
<tr>
<td><strong>italic</strong></td>
<td>The italic attribute is used for emphasis or to indicate a related document. (See the <strong>Installation Guide</strong> for more information.)</td>
</tr>
</tbody>
</table>

Double quote marks

Double quote marks enclose references to other sections within the document.

<variable>

In command descriptions, angle brackets represent variables. You must substitute a value for command line arguments that are enclosed in angle brackets.

[ optional ]
[ <optional> ]

Square brackets enclose optional keywords or <variables> in a command line description. Optionally enter the keyword or <variable> that is enclosed in square brackets, if it is necessary or desirable to complete the task.

[ a | b | c ]
[<a> | <b> | <c>]  

Square brackets enclose optional alternate keywords or variables in a command line description. Choose one command line argument enclosed within the braces, if desired. Choices are separated by vertical (OR) bars.

{ a | b | c }  
{<a> | <b> | <c>}  

Braces enclose required alternate keywords or <variables> in a command line description. You must choose one command line argument enclosed within the braces. Choices are separated by vertical (OR) bars.
Support Contacts

If you encounter a problem while installing, registering, or operating this product, refer to the documentation. If you cannot resolve the issue, contact your supplier or Gemalto Customer Support.

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

Customer Support Portal

The Customer Support Portal, at https://supportportal.gemalto.com, is a where you can find solutions for most common problems. The Customer Support Portal is a comprehensive, fully searchable database of support resources, including software and firmware downloads, release notes listing known problems and workarounds, a knowledge base, FAQs, product documentation, technical notes, and more. You can also use the portal to create and manage support cases.

NOTE: You require an account to access the Customer Support Portal. To create a new account, go to the portal and click on the REGISTER link.

Telephone Support

If you have an urgent problem, or cannot access the Customer Support Portal, you can contact Gemalto Customer Support by telephone at +1 410-931-7520. Additional local telephone support numbers are listed on the support portal.

Email Support

You can also contact technical support by email at technical.support@gemalto.com.
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. SafeNet provides access to the Luna EKM, which includes the EKM Provider Library. You must configure the EKM provider option to use the HSM device with SQL server.

This document provides low-level details of how the SafeNet Luna Hardware Security Module (HSM) or HSMoD service 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.

Using SafeNet HSMs to secure the Microsoft SQL Server key provides the following benefits:

- Secure generation, storage and protection of the Identity signing private key on FIPS 140-2 level 3 validated hardware*.
- Full life cycle management of the keys.
- HSM audit trail.
- Load balancing and fail-over by clustering the HSMs.
- Take advantage of cloud services with confidence.
- Significant performance improvements by off-loading cryptographic operations from application servers

*validation for HSMoD services in progress.
**HSMoD services do not have access to the secure audit trail.

Third Party Application Details
This integration uses the following third party applications:

- Microsoft SQL Server

Microsoft SQL Server
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 the HSM in a Microsoft SQL configuration:

**Microsoft SQL Server High Availability (Always On)**

The Always On Availability group feature is a high-availability and disaster recovery solution that provides an enterprise-level alternative to database mirroring. 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.

**NOTE:** Support for Higher Length Asymmetric Keys on Microsoft SQL Server and Microsoft SQL Server HA configurations. 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.

For Luna EKM setup, contact Customer support.

DOC ID for EKM 1.3 is KB0014957.

DOC ID for EKM 1.4 is KB0016274.
Microsoft SQL Server Always Encrypted

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 will not appear in plaintext. 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:

![Diagram of SQL Server Always Encrypted feature](Image)

**NOTE:** SQL Server Always Encrypted is supported in both FIPS and NON-FIPS mode.

**Supported Platforms**

List of the platforms which are tested with the following HSMs:

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

This integration is supported with SafeNet Luna HSM on the following operating systems:

<table>
<thead>
<tr>
<th>Platforms Tested</th>
<th>EKM Software Version</th>
<th>Microsoft SQL Server</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows Server 2016</td>
<td>EKM v1.4</td>
<td>Microsoft SQL Server 2017</td>
</tr>
<tr>
<td>Windows Server 2012 R2</td>
<td></td>
<td>Microsoft SQL Server 2016</td>
</tr>
</tbody>
</table>

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

This integration is supported/verified with SafeNet DPoD on the following operating systems:

<table>
<thead>
<tr>
<th>Platforms Tested</th>
<th>EKM Software Version</th>
<th>Microsoft SQL Server</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows Server 2016</td>
<td>EKM v1.4</td>
<td>Microsoft SQL Server 2017</td>
</tr>
</tbody>
</table>

**Prerequisites**

Before proceeding with the integration, ensure you have completed the following. Complete the section *Configure the SafeNet Luna HSM* or *Provision your HSM on Demand Service* depending on the integration you are completing.

Follow the *SafeNet Luna Network HSM Product Documentation* for detailed steps for creating the NTLS connection, initializing the partitions, and initializing the Security Officer, Crypto Officer, and Crypto User roles.

**NOTE**: SQL Server does not implement FIPS 186-4. As a result, you cannot generate RSA keys when the HSM in FIPS Mode. To use Asymmetric Key Encryption use the HSM in NON-FIPS mode.

**Configure the SafeNet Luna HSM**

If you are using a SafeNet Luna HSM, ensure the following:

1. Ensure the HSM is set up, initialized, provisioned and ready for deployment.
2. Create a partition on the SafeNet Luna HSM for use with Microsoft SQL Server.
3. Register a client for the system and assign the client to partition to create an NTLS connection. Initialize the Crypto Officer and Crypto User roles for registered partition.
4. Ensure that each partition is successfully registered and configured. The command to see the registered partitions is:

```
# /usr/safenet/lunaclient/bin/lunacm
LunaCM v7.1.0-379. Copyright (c) 2006-2017 SafeNet.
```

Available HSMs:

<table>
<thead>
<tr>
<th>Slot Id</th>
<th>Label</th>
<th>Serial Number</th>
<th>Model</th>
<th>Firmware Version</th>
<th>Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>ms-sql</td>
<td>1238712343066</td>
<td>LunaSA 7.1.0</td>
<td>7.1.0</td>
<td>Luna User Partition With SO (PED) Key Export</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Net Token Slot</td>
</tr>
</tbody>
</table>

**Provision your HSM on Demand Service**

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

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

> **Security Officer (SO)** - responsible for setting the partition policies and for creating the Crypto Officer.

> **Crypto Officer (CO)** - responsible for creating, modifying and deleting crypto objects within the partition. The CO can use the crypto objects and create an optional, limited-capability role called Crypto User that can use the crypto objects but cannot modify them.

> **Crypto User (CU)** - optional role that can use crypto objects while performing cryptographic operations.

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

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

**Constraints on HSMoD Services**

Please take the following limitations into consideration when provisioning your HSMoD services:

**HSM on Demand Service in FIPS mode**

HSMoD services operate in a FIPS and non-FIPS mode. If your organization requires non-FIPS algorithms for your operations, ensure you enable the **Allow non-FIPS approved algorithms** check box when configuring your HSM on Demand service. The FIPS mode is enabled by default.
Refer to the *Mechanism List* in the SDK Reference Guide for more information about available FIPS and non-FIPS algorithms.

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

**Access to 32-bit KSP Library**
If configuring SQL Server Always Encrypted (AE), the 32-bit KSP library and utility is not available with all versions of the HSM on Demand Windows service client. If you do not have the 32-bit KSP library and would like to configure SQL Server Always Encrypted, please contact Gemalto Customer Support for access to the library and utility.

**Set up Luna EKM**
Install Luna EKM on the host system. A Windows-based installation program, lunaEKMconfig is provided to assist with the EKM installation.

LunaEKM includes a command line configuration utility `lunaEKMConfig` that is used to register the Luna EKM. This command line utility is available in the LunaEKM installation folder. LunaEKMconfig provides command to register slots, view slots, and to configure log settings.

Run the following commands, available in LunaEKMConfig.

1. Register the slot of LunaEKM to use.
   ```
   RegisterSlot
   ```
2. View List of the Slots/HSM configured with this client.
   ```
   ViewSlots
   ```
3. Configure log settings for LunaEKM.
   ```
   LogSettings
   LogLevel (NONE=0, INFO=1, DEBUG=2): <LogLevel>
   LogFile name: <Name and location of LogFile>
   ```

**Set up SQL Server**
Install SQL Server on the target machine to complete the integration process. If you are configuring a high availability (Always ON) SQL server group, the SQL Server must be installed on all nodes and all nodes must have access to WFCS.

Refer to the *Microsoft SQL Server Documentation* for detailed installation procedures.
CHAPTER 2: Integrating SafeNet HSM with SQL Server

This document contains detailed instructions and procedures to integrate Microsoft SQL Server with a SafeNet Luna HSM or HSM on Demand service. This integration contains the following topics:

- Enabling the EKM Provider option
- Creating and Registering the Luna EKM Provider
- Setting up the CREDENTIAL for Luna EKM Provider
- Using the Luna EKM Provider Option
- Enabling Transparent Database Encryption using Asymmetric key on SafeNet HSM
- Rotating Keys for Transparent Database Encryption
- Migrating TDE from SQL EKM to Luna EKM
- Using Extensible Key Management on a SQL Server Failover Cluster

Enabling the EKM Provider option

Use the `sp_configure` command to enable the EKM Provider option.

**To enable the EKM provider option**

1. Open the SQL Server Management Studio.
2. Connect to the SQL Server.
3. Open a query window, and execute the following:
   ```
   sp_configure 'show advanced', 1
   GO
   RECONFIGURE
   GO
   sp_configure 'EKM provider enabled', 1
   GO
   RECONFIGURE
   GO
   ```

   **NOTE:** The `sp_configure` command is supported on Enterprise, Developer, and Evaluation editions of SQL server. If you execute the command on an alternative version, you will receive an error.
Creating and Registering the Luna EKM Provider

Setup the Luna EKM provider. Install the Luna EKM Software and register it for use with SQL Server.

To create and register the Luna EKM Provider

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

   CREATE CRYPTOGRAPHIC PROVIDER <Name of Cryptographic Provider>
   FROM FILE = '<Location of Luna EKM Provider Library>'

   where <Name of Cryptographic Provider> can be any user defined unique name.

4. Verify the list of EKM providers:

   SELECT [provider_id], [name], [guid], [version], [dll_path], [is_enabled]
   FROM [model].[sys].[cryptographic_providers]

5. Verify 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 the CREDENTIAL for Luna EKM Provider

The next step is to create a CREDENTIAL for the Luna EKM Provider. You must map the CREDENTIAL to the SQL User or login to use the Luna EKM Provider option.

To setup the CREDENTIAL for Luna EKM Provider

1. Open a query window, and execute 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 you are using a PED based SafeNet PCIe HSM the value for the secret is a single space character “ “.

2. Map the LunaEKMCred with SQL User or Login:

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

   **NOTE:** The EKM session must be reopened if the user changes the HSM service, the client machine is deleted from the service, or the machine suffers a network disconnection.

Using the Luna EKM Provider Option

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.

The 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.

Follow the below steps to exercise the cryptographic capabilities of the SafeNet Luna HSM from the SQL Server:

Creating Symmetric Keys on SafeNet HSM

The following examples use AES algorithms for the symmetric key operation. To test other algorithms, substitute AES_256 with an alternate algorithm tag, such as RC2, or Triple_DES.

**To create the symmetric key using the Luna EKM Provider**

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

   ```sql
   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 HSM, it can be used or referred to 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. Using this key name will use the key on the HSM for encrypt and decrypt operations.
To view symmetric key using the Luna EKM Provider

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

```sql
SELECT * FROM [master].[sys].[symmetric_keys]
```

To encrypt a database table with symmetric keys using the Luna EKM Provider

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'))
```

To decrypt a database table with symmetric keys using the Luna EKM Provider

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
```

To drop symmetric keys using the Luna EKM Provider

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

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

Creating Asymmetric Keys on SafeNet HSM

The 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

The following examples use RSA_2048 algorithms for asymmetric key operation. To test other algorithms, substitute RSA-2048 with an alternate algorithm tag, such as RSA_512.

To create the asymmetric key using the Luna EKM Provider

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

```sql
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 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. Using this key name will use the key on the HSM for encrypt and decrypt operations.

To view asymmetric keys using the Luna EKM Provider

1. Execute the following:

   SELECT * FROM [master].[sys].[asymmetric_keys]

To encrypt a database table with asymmetric keys using the Luna EKM Provider

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'))

To decrypt a database table with asymmetric keys using the Luna EKM Provider

You can decrypt tables in the database using the Luna EKM Provider key stored on the SafeNet HSM.

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

To drop asymmetric keys using the Luna EKM Provider

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

   DROP ASYMMETRIC KEY SQL_EKM_RSA_2048_Key REMOVE PROVIDER KEY

Creating Symmetric Key Encrypted by Asymmetric Key on SafeNet HSM

You can encrypt a symmetric key using an asymmetric key. This increases the security of the symmetric key.

To create a symmetric key encrypted by an asymmetric key

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;
CHAPTER 2: Integrating SafeNet HSM with SQL Server

NOTE: SQL_EKM_RSA_2048_Key is an existing asymmetric key on the SafeNet HSM. For more information about generating the asymmetric key see Creating Asymmetric Keys on SafeNet HSM.

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;
```

NOTE: For Microsoft SQL Server 2017, apply the patch as described in the Troubleshooting Problem – 3.

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

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

4. Encrypt the table data using the symmetric key.

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

5. Decrypt the data using the symmetric key.

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

6. Close the symmetric key.

```
CLOSE SYMMETRIC KEY key1
```

Enabling Transparent Database Encryption using Asymmetric key on SafeNet HSM

You can enable Transparent Data Encryption (TDE) using an asymmetric key stored on a SafeNet HSM. Execute the following from the SQL query window.

NOTE: Database encryption operations cannot be executed on 'master', 'model', 'tempdb', 'msdb', or 'resource' databases.

To enable TDE using asymmetric key on SafeNet HSM

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',
```

Microsoft SQL Server: INTEGRATION GUIDE
007-011108-001, Rev. AK, March 2019 Copyright © 2019 Gemalto
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 recently created asymmetric key.

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

4. Map the credential created above to the recently created login.

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

5. Create a Database Encryption Key.

   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. Query the status of database encryption and the completion percentage.

   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

---

Rotating Keys for Transparent Database Encryption

We recommend updating your TDE security keys regularly by rotating the available symmetric and asymmetric encryption keys. Execute the following from the SQL query window.
To rotate keys for TDE

1. Generate an asymmetric key using the Luna EKM Provider.
   
   Use master;
   
   ```sql
   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.
   
   ```sql
   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 recently created asymmetric key.
   
   ```sql
   CREATE LOGIN <Name of login>
   FROM ASYMMETRIC KEY SQL_EKM_RSA_2048_Key_TDE_Rot;
   ```

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

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

6. Execute the following command to query the status of database encryption, the status of TDE key change, and the tablespace encryptions percentage of 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

Migrating TDE from SQL EKM to Luna EKM

Previously, the database master key was generated in SQL and encrypted using a certificate or asymmetric key. Now, you can generate a new database master key encrypted by an asymmetric key generated on a SafeNet HSM.

This example uses the database name <AdventureWorks>.

To migrate TDE from SQL EKM to Luna EKM

1. Decrypt the database (AdventureWorks)

   USE master;
   ALTER DATABASE AdventureWorks
   SET ENCRYPTION OFF;
   GO

2. Backup the database and transaction logs. When the backup completes, restart the SQL database.

3. Create an asymmetric key.

   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 recently created asymmetric key.

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

6. Map the CREDENTIAL to the recently created login.

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

7. Create or Regenerate the Database Encryption Key (DEK).

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

   Regenerate:
   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. Query the status of database encryption and its completion percentage.
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
If the Query executed successfully, a success message displays. The integration between SQL server and the SafeNet Luna HSM or HSM on Demand service is complete.

Using Extensible Key Management on a SQL Server Failover Cluster

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

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 the SafeNet Luna Network HSM client or HSM on Demand service client on both the nodes.

3. Configure and setup the HSM on both the nodes and register the same partition or HSMoD service on both nodes in the SQL Server Cluster.

4. Install the 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 encryption keys using the Luna EKM provider on the first node.
9. Create a table and encrypt a column with the Luna EKM key on the first node.
10. Shutdown the first node.
11. Login to the second node and decrypt the data encrypted on the first node.
12. If the data decrypts successfully, Extensible Key Management (EKM) using Luna EKM is operating correctly on the SQL Server cluster.
CHAPTER 3: Integrating SafeNet HSM with SQL Server High Availability Group

To integrate the SafeNet HSM with SQL Server, you must set up and configure the Luna Extensible Key Management (EKM) Provider and enable the EKM provider in the SQL server. The EKM feature is available on the Enterprise, Developer, and Evaluation editions of the SQL server. EKM is disabled by default.

You can set up your SQL server in a High Availability configuration for failover support. Luna Client and Luna EKM must be setup on all SQL Server cluster nodes that will be added to the “Always On” availability group. All nodes must be registered with the same partition of SafeNet Luna HSM or the same service client on an HSMoD service.

This integration contains the following topics:

- Enabling the EKM Provider Option
- Creating and Registering the Luna EKM Provider
- Setting up the CREDENTIAL for Luna EKM Provider
- Creating the Always On Availability Group
- Creating the Encryption Keys for Availability Group Database
- Enabling Transparent Database Encryption using Asymmetric key on SafeNet HSM
- Adding the Encrypted Database to the Availability Group
- Rotating Keys for Transparent Database Encryption

Enabling the EKM Provider Option

Use the `sp_configure` command to enable the EKM Provider option on all nodes in the high availability configuration.

**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 execute the following:

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

NOTE: The `sp_configure` command is supported on Enterprise, Developer, and Evaluation editions of SQL Server. If you execute the command on an alternative version, you will receive an error.

Creating and Registering the Luna EKM Provider

Setup the Luna EKM provider. Install the Luna EKM Software and register it for use with SQL Server on all nodes in the high availability configuration.

To create and register the Luna EKM Provider

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

   ```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. 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 the CREDENTIAL for Luna EKM Provider

Create a CREDENTIAL for the Luna EKM Provider and map the CREDENTIAL to the SQL User or Login to use the Luna EKM Provider option on all nodes in the High Availability configuration.
To setup the CREDENTIAL for Luna EKM Provider:

1. Open a query window, and execute 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. Map the LunaEKMCred to the SQL User or Login:

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

   **NOTE:** We recommend using a domain user on all SQL Server nodes. The EKM session must be reopened if the user changes the HSM service, the client machine is deleted from the service, or the machine suffers a network disconnection.

Creating the Always On Availability Group

Create the Always On Availability group and configure the nodes in the cluster to communicate with each other. For detailed installation procedures refer to the Microsoft Online Documentation for creating the Always on Availability group.

To create the Always On Availability Group

1. Open the Microsoft SQL Server management Studio on the primary node.
2. Create a database.
3. Backup the database to a shared network location that is accessible by all of the SQL Server nodes.
4. Open the Always On Availability Group Creation wizard and create an Always On Availability group for the cluster configuration.
After the successful creation of the group, the dashboard displays all the participating nodes. An example of a dashboard is shown below. For demonstration purposes, two nodes were added; primary and secondary.

Creating the Encryption Keys for Availability Group Database

You can use the Luna EKM provider to create/drop symmetric and asymmetric keys to/from the HSM and can perform encryption/decryption using these keys.

Creating Symmetric Keys on SafeNet HSM

The following types of symmetric key can be created on SafeNet HSM from the SQL Server:

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

* Depreciated in SQL Server 2012.

The following examples use AES algorithms for symmetric key operation. To test other algorithms, substitute AES_256 with an alternate algorithm tag, such as RC2, or Triple_DES.

To create the symmetric key using the Luna EKM Provider

1. Open the SMS on the primary node.
2. Execute the following command from the SQL query window:
   ```sql
   USE HSMDB;
   ```
3. Execute the following command from the SQL query window:
   ```sql
   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 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. Using this key name will use the key on the HSM for encrypt and decrypt operations.

To view symmetric keys using the Luna EKM Provider
1. Execute the following command from the SQL query window:
   SELECT * FROM [hsmdb].[sys].[symmetric_keys]

To encrypt a database table with symmetric keys using the Luna EKM Provider
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'))

To decrypt a database table with symmetric keys using the Luna EKM Provider
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 HSM
The following types of asymmetric key can be created on SafeNet HSM from the SQL Server:
> RSA_512
> RSA_1024
> RSA_2048
The following examples use RSA_2048 algorithms for asymmetric key operation. To test other algorithms, substitute RSA_2048 with an alternate algorithm tag, such as RSA_512.
To create the asymmetric key using the Lune EKM Provider

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

```sql
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:** When you create a key on the SafeNet Luna HSM or HSMoD service, you can use the key in SQL commands by identifying the key by its object label. In the preceding example `SQL_EKM_RSA_2048_Key` is the unique name of the key in the SQL server. Using this key we can perform crypto operation.

To view asymmetric keys using the Lune EKM Provider

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

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

To encrypt a database table with asymmetric keys using the Lune EKM Provider

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

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

To decrypt a database table with asymmetric keys using the Lune EKM Provider

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

```sql
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 HSM

You can encrypt the symmetric keys using an asymmetric key. This increases the security of the symmetric key.

To create a symmetric key encrypted by an asymmetric key

1. Open the SMS on the primary node.
2. Execute the following command from SQL query window:

```sql
Create SYMMETRIC KEY key1
WITH ALGORITHM = AES_256
```
ENCRIPTION BY Asymmetric Key SQL_EKM_RSA_2048_Key;
where “SQL_EKM_RSA_2048_Key” is an existing asymmetric key.

3. 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;

NOTE: For Microsoft SQL Server 2017, apply the patch as described in the Troubleshooting Problem – 3

4. Encrypt the data using the key1:

INSERT INTO dbo.test
    values ( 3,'some text',
    Encryptbykey(KEY_GUID('Key1'), 'text to be encrypted'))

5. Decrypt the data using the key1:

SELECT id,name,CONVERT(varchar(MAX),
    DecryptByKey(data))
FROM dbo.test where id =3

6. Close the symmetric key.

CLOSE SYMMETRIC KEY key1

7. Now execute the above steps (3-6) on secondary replica and verify that the output is same as primary replica.

Enabling Transparent Database Encryption using Asymmetric key on SafeNet HSM

You can enable Transparent Data Encryption (TDE) using an asymmetric key stored on a SafeNet HSM.

NOTE: Database encryption operations cannot be executed on 'master', 'model', 'tempdb', 'msdb', or 'resource' databases.

To enable TDE using asymmetric key on SafeNet HSM

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 the 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.
   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 recently created asymmetric key.
   CREATE LOGIN <Name of login>
   FROM ASYMMETRIC KEY SQL_EKM_RSA_2048_KEY_TDE;

5. Map the CREDENTIAL to the recently created Login.
   ALTER LOGIN <Name of Login>
   ADD CREDENTIAL <Name of credential>;

6. Execute the above steps (2-5) for all secondary nodes.
   NOTE: Repeating the procedure is required for all nodes in the database because the
   TDE encryption key, CREDENTIAL, and Login, are objects in the master database and
   are not replicated by including the node in the Availability Groups.

7. Create a Database Encryption Key on the primary node.
   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:
   ALTER DATABASE TDE
   SET ENCRYPTION ON;

9. Query the status of database encryption and its completion percentage.
   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
Adding the Encrypted Database to the Availability Group

Before adding the already encrypted database into the availability group, backup the encrypted database to a network location that is accessible by all secondary nodes.

To add the encrypted database to the availability group

1. Open the SMS on the primary node.
2. Add the database (e.g. TDE) into the availability group (e.g. AGroup).
   
   ```
   use master;
   ALTER AVAILABILITY GROUP AGroup ADD DATABASE tde;
   GO
   ```
   
   This command adds the database to the Availability Group, but it is not yet available on the secondary node. To access the encrypted database from the secondary node you need to synchronize the databases by restoring the database on the second node.

3. Restore the database on the secondary node. Restore the database from the location where you have stored the encrypted database with the “RESTORE WITH NORECOVERY” parameter.

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

5. Query the status of database encryption and its completion percentage on the 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
   ```

Rotating Keys for Transparent Database Encryption

We recommend updating your TDE security keys regularly by rotating the available symmetric and asymmetric encryption keys.

To rotate keys for TDE

1. Create an asymmetric key using the Luna EKM Provider on the primary node.
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 the same asymmetric key using the Luna EKM Provider on secondary node.

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 recently created asymmetric key.

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

5. Map the CREDENTIAL to the recently created Login.

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

6. Execute steps 2-5 for all secondary nodes.

7. Enable TDE 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. Query the status of database encryption, TDE key change and its completion percentage.

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

This completes the integration of Microsoft SQL Server High Availability with a SafeNet Luna Network HSM or an HSM on Demand service.
CHAPTER 4: Integrating SafeNet HSM with SQL Server Always Encrypted

This document contains detailed instructions to integrate Microsoft SQL Server Always Encrypted with a SafeNet Luna HSM. This integration contains the following topics:

- Configuring the SafeNet KSP
- Generating the Column Master Key
- Generating the Column Encryption Key
- Implementing Always Encrypted
- Viewing Always Encrypted Data

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

**Note:** 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 or HSM on Demand service in FIPS mode, you have to make the following change in the configuration file:

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

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

## Configuring the SafeNet KSP

Register the SafeNet Luna Key Storage Provider (KSP) on the target machine to generate the column master key and encryption key on the SafeNet Luna HSM.

**To configure the SafeNet KSP**

1. Navigate to the **32-bit Luna KSP** directory.
   
   `<Luna Client Installation Directory>\win32\KSP`

   **For Example:** `cd "C:\Program Files\SafeNet\LunaClient\win32\KSP"

   **NOTE:** If using an HSM on Demand Service the KSP directory is available in the service client.

2. Run the KSPConfig.exe (KSP configuration wizard) utility to register the SafeNet Luna KSP through a GUI. The general form of command is:
CHAPTER 4: Integrating SafeNet HSM with SQL Server Always Encrypted

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 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.
   a. Open the Register for User drop-down menu and select the User. Open the Domain drop-down and select your domain.
   b. Open the Available Slots drop-down and select the partition.
   c. Enter the partition password in Slot Password field.
   d. Click Register Slot to register the slot for Domain\User. On successful registration, a message "The slot was successfully and securely registered" displays.
7. Double-click Register HSM Slots on the left side of the pane.
   a. Open the Register for User drop-down menu and select NT_AUTHORITY. Open the Domain drop-down and select Domain.
   b. Open the Available Slots drop-down and select the partition.
   c. Enter the partition password in Slot Password field.
   d. Click Register Slot to register the slot for NT_AUTHORITY\SYSTEM. On successful registration, a message "The slot was successfully and securely registered" displays.

NOTE: The partition has been registered for both users, despite only one entry appearing for the <slot_label> in the Registered Slots section of the KSP interface.

Generating the Column Master Key

You require a Column Master Key to configure Always Encrypted.

To generate the column master key
1. Connect to the database using SQL Server Management Studio from a client machine.
2. In Object Explorer, navigate to Databases > Test > Security.
   
   NOTE: Test is the sample database created for demonstration purpose.
3. Expand the Always Encrypted Keys folder to display its two subfolders:
   - Column Master Keys
   - Column Encryption Keys
4. Right-click on **Column Master Keys** and select **New Column Master Key**. The **New Column Master Key** wizard displays.

5. Enter a name for the Column Master Key Pair in the **Name** field.

6. Open the **Key Store** drop-down menu and select **Key Storage Provider (CNG)**.

7. Open the **Select a provider** drop-down menu and select **SafeNet Key Storage Provider**.
8. Click **Generate Key**.
9. The key pair (column master key) generates on the SafeNet HSM partition. Execute **partition content** in lunacm to verify the keys exist.

```
C:\Program Files\SafeNet\LunaClient\win32> lunacm.exe
lunacm.exe (32-bit) v7.2.0-220. Copyright (c) 2018 SafeNet. All rights reserved.

Available HSMs:
Slot Id -> 0
Label -> SQL-AE
Serial Number -> 1213475834468
Model -> LunaSA 7.2.0
Firmware Version -> 7.2.0
Configuration -> Luna User Partition With SO (PW) Signing With Cloning Mode
Slot Description -> Net Token Slot

Current Slot Id: 0
```

```
lunacm> role login -n co -p userpin1
Command Result : No Error
lunacm> partition contents
The 'Crypto Officer' is currently logged in. Looking for objects accessible to the 'Crypto Officer'.
Object list:
Label: Always-Encrypted-Auto1
Handle: 15012
Object Type: Private Key
Object UID: d2170a0970000009e0f30700
Label: Always-Encrypted-Auto1
Handle: 15009
Object Type: Public Key
Object UID: d1170a0870000009e0f30700
Number of objects: 2
```

### Generating the Column Encryption Key

You require a Column Encryption Key to configure Always Encrypted.

**To generate the column encryption key**

1. In the Object Explorer, navigate to **Databases** > **Test** > **Security**.
2. Expand the **Always Encrypted Keys** folder to display its two subfolders:
   - Column Master Keys
   - Column Encryption Keys
3. Right-click on **Column Encryption Keys** and select **New Column Encryption Key**. The **New Column Encryption Key** wizard displays.

![](image)

4. Enter a name for the Column Encryption Key in the **Name** field. Click the **Column Master Key** drop-down menu and select **LUNAKEY**.

5. Click **OK**.
6. The key generates and is stored in the SQL Server Instance where Always Encrypted is implemented.

Implementing Always Encrypted

When you have configured the Column Master Key and Column Encryption Key you can implement Always Encrypted on the SQL server.

To implement Always Encrypted

1. Create the table **Employee** with fields in database **Test**.

   ```sql
   use Test;
   Create Table Employee(
     id numeric(10),
     name varchar (50),
     data varchar (max),
   );
   ```

2. Insert some values into the table.

   ```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. Encrypt the Employee details. Right-click the Employee table and select Encrypt Columns. The **Always Encrypted** wizard displays.

5. The introduction page displays. Click **Next**.
6. The **Column Selection** page allows you to select the columns to encrypt and the encryption type:

   - Randomized
   - Deterministic

For the Employee table, set the Data column to be **Randomized**, and the id column to be **Deterministic**. Click **Next**.

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

8. On the Run Settings page, select Proceed to finish now. Click **Next**.

9. Verify the details on **Summary** page. Click **Finish** to complete the encryption process.
10. Once the process completes, click **Close**.

11. View the table contents.

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

   The Columns data and id now appear in encrypted form.

**Viewing Always Encrypted Data**

Once you have configured SQL Server Always Encrypted you may need to access the encrypted data.

**To view Always Encrypted data**

1. Select the **Query** Menu tab in SSMS. Point to **Connection**.
2. Click **Change Connection**. The **Connect to Database Engine** dialog displays.
3. Click **Options** and select the **Additional Connection Parameters** tab.
5. On this tab, enter the following value:
   Column Encryption Setting=Enabled

6. Click Connect.

7. Run the following query:

   use Test;
   Select * from dbo.Employee;

8. Pop-up screen will prompt for Parameterization for Always Encrypted. Click Enable.

   Now unencrypted data of table displays.

   This completes the configuration of SQL Server Always Encrypted. The column master key is secured in a SafeNet Luna HSM or HSMoD service and the column encryption key is encrypted using the securely stored master key.
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
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. This should resolves the above problem.

Problem – 3
Unable to open Symmetric key which is encrypted by Asymmetric Key in Microsoft SQL Server 2017.

Error:
Msg 15466, Level 16, State 28, Line 1
An error occurred during decryption.

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

- For SQL Server 2017: