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

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

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

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

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

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

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

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

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

© 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. AH
Release Date: August 2018
Rotating Keys for Transparent Database Encryption ................................................................. 34

4 SafeNet Luna HSM Integration with SQL Server Always Encrypted ............................... 36
   Configuring the SafeNet KSP ........................................................................................................ 36
   Generating the Column Master Key ........................................................................................... 37
   Generating the Column Encryption Key ..................................................................................... 40
   Implementing Always Encrypted .................................................................................................. 42
   Viewing Always Encrypted Data .................................................................................................. 44

5 Troubleshooting Tips .................................................................................................................. 46
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. Contains important or helpful information.

Cautions

Cautions are used to alert you to important information that may help prevent unexpected results or data loss. These elements use the following format:

CAUTION: Exercise caution. Caution alerts contain important information that may help prevent unexpected results or data loss.

Warnings

Warnings are used to alert you to the potential for catastrophic data loss or personal injury. These elements use the following format:

WARNING: Be extremely careful and obey all safety and security measures. In this situation you might do something that could result in catastrophic data loss or personal injury.
## Command Syntax and Typeface Conventions

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

<table>
<thead>
<tr>
<th>Contact Method</th>
<th>Contact Information</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Address</strong></td>
<td>Gemalto</td>
</tr>
<tr>
<td></td>
<td>4690 Millennium Drive</td>
</tr>
<tr>
<td></td>
<td>Belcamp, Maryland 21017, USA</td>
</tr>
<tr>
<td><strong>Phone</strong></td>
<td>US</td>
</tr>
<tr>
<td></td>
<td>1-800-545-6608</td>
</tr>
<tr>
<td></td>
<td>International</td>
</tr>
<tr>
<td></td>
<td>1-410-931-7520</td>
</tr>
<tr>
<td><strong>Technical Support</strong></td>
<td><a href="https://supportportal.gemalto.com">https://supportportal.gemalto.com</a></td>
</tr>
<tr>
<td><strong>Customer Portal</strong></td>
<td>Existing customers with a Technical Support Customer Portal account can log in to</td>
</tr>
<tr>
<td></td>
<td>manage incidents, get the latest software upgrades, and access the Gemalto Knowledge</td>
</tr>
<tr>
<td></td>
<td>Base.</td>
</tr>
</tbody>
</table>
Overview

SQL Server allows for use of HSM devices to store keys and perform cryptographic operations such as key creation, key deletion, encryption, and decryption through use of the Extensible Key Management (EKM) feature. You can encrypt data using encrypting keys that are only accessible by the database users on the external EKM/HSM module.

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.

This document provides low-level details of how the SafeNet Luna Hardware Security Modules (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.

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

*FIPS 140-2 validation in progress for HSMoD services

The integration entails the following:

- Install and configure SafeNet Luna HSM or HSMoD service.
- Install and configure SafeNet Luna EKM.
Third 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 the HSM in a Microsoft SQL configuration:

![Database Master Key Diagram](diagram.png)

About Microsoft SQL Server High Availability (Always On)

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 maximize 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.

About 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 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:

---

**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.

![Support for Higher Length Asymmetric Keys](image)

**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**

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>
</tr>
</thead>
<tbody>
<tr>
<td>Windows Server 2016</td>
<td>EKM v1.4</td>
</tr>
<tr>
<td>Windows Server 2012 R2</td>
<td></td>
</tr>
</tbody>
</table>

**NOTE:** This integration is tested with Luna Clients in HA and FIPS Mode.

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 only the services that you need.

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

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

**Prerequisites**

Before you proceed with the integration, ensure that you complete Configuring the SafeNet Luna Network HSM or Provision your HSM on Demand service as per the integration requirement.

**Configuring SafeNet Luna HSM**

Before you get started ensure the following:

1. Ensure the HSM is setup, initialized, provisioned and ready for deployment. Refer to the HSM product documentation for help.
2. Create a partition on the HSM that will be later used by Microsoft SQL.
3. If using a SafeNet Luna Network HSM, register a client for the system and assign the client to the partition to create an NTLS connection. Initialize Crypto Officer and Crypto User roles for the registered partition.
4. Ensure that the partition is successfully registered and configured. The command to see the registered partitions is:
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>
<th>Slot Description</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 With Cloning Mode</td>
<td>Net Token Slot</td>
</tr>
</tbody>
</table>

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

### Provision your HSM on Demand Service

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

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

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

**NOTE:** Refer to the “SafeNet Data Protection on Demand Application Owner Quick Start Guide” for procedural information on configuring the HSM on Demand service and create a service client. The HSM on Demand Service client package is a zip file that contains system information needed to connect your client machine to an existing HSM on Demand service.

### Constraints on HSM on Demand Services

Please consider the following if integrating an HSMoD service with Microsoft SQL Server:
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.

Using SafeNet HSM in FIPS Mode

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

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.

Setup Luna EKM

A Windows-based installation program is provided to make the Luna EKM installation quick and easy - lunaEKMconfig. 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 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>"`
Setup 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.

We recommend you access the *Microsoft SQL Server* online documentation for detailed installation procedures.
SafeNet HSM Integration 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
- Using the Luna EKM Provider Option
- Creating Symmetric Key Encrypted by Asymmetric Key on SafeNet HSM
- Enabling Transparent Database Encryption using Asymmetric key on SafeNet HSM
- Rotating Keys for Transparent Database Encryption

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.

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

To setup the Luna EKM provider, you must 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:
   
   ```sql
   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:
      
      ```sql
      SELECT [provider_id], [name], [guid], [version], [dll_path], [is_enabled]
      FROM [model].[sys].[cryptographic_providers]
      ```
      
   5. Verify 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

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:

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

Creating Symmetric Keys on SafeNet 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.

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. Execute the following command from the SQL query window:

   CREATE SYMMETRIC KEY SQL_EKM_AES_256_Key
   FROM Provider LunaEKMPProvider
   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 key using the Luna EKM Provider
You can view symmetric keys by executing the following command from the SQL query window.

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

To encrypt with symmetric keys using the Luna EKM Provider
You can encrypt tables in the database using the Luna EKM Provider key stored on the SafeNet HSM.

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

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

3. 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 with symmetric 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:

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

To drop symmetric keys using the Luna EKM Provider
You can drop symmetric keys using the Luna EKM Provider. The following command drops the key from the SQL Server and the SafeNet HSM.

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
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:

```
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**

You can view asymmetric keys by executing the following command from the SQL Query window.

1. Execute the following:

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

**To encrypt with asymmetric keys using the Luna EKM Provider**

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

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 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**

You can drop asymmetric keys using the Luna EKM Provider. The following example command will drop the keys from both the SQL server and the SafeNet HSM.

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 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. Execute the following command from SQL query window:

```sql
CREATE SYMMETRIC KEY key1
WITH ALGORITHM = AES_256
ENCRYPTION BY Asymmetric Key SQL_EKM_RSA_2048_Key;
```

**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:

```sql
OPEN SYMMETRIC KEY key1 DECRYPTION BY Asymmetric Key SQL_EKM_RSA_2048_Key;
```

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

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

4. Encrypt the table data using the symmetric key.

```sql
INSERT INTO dbo.test
VALUES (1,'some text',
EncryptByKey(KEY_GUID('key1'),'text to be encrypted'))
```

5. Decrypt the data using the symmetric key.

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

6. Close the symmetric key.

```sql
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.

**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.

   ```sql
   USE master;
   CREATE ASYMMETRIC KEY SQL_EKM_RSA_2048_Key_TDE
   FROM Provider LunaEKMProvider
   WITH ALGORITHM = RSA_2048,
   ```
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.

To rotate keys for TDE

1. Generate an asymmetric key using the 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 IDENTIFY='<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_Rot;

4. Map the credential recently created login.
   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. Execute the following command to query the status of database encryption, the status of TDE key change,
   and the tablespace encryptions percentage of 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

Migrating TDE from SQL EKM to Luna EKM

If you enable Transparent Data Encryption (TDE) for any database using the SQL EKM and desire to migrate
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 generate 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)
   ```sql
   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.
   ```sql
   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.
   ```sql
   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.
   ```sql
   CREATE LOGIN <Name of login>
   FROM ASYMMETRIC KEY SQL_EKM_RSA_2048_Key_AW;
   ```
6. Map the CREDENTIAL to the recently created login.
   ```sql
   ALTER LOGIN <Name of Login>
   ADD CREDENTIAL <Name of credential>;
   ```
7. Create or Regenerate the Database Encryption Key (DEK).
   ```sql
   USE AdventureWorks;
   CREATE DATABASE ENCRYPTION KEY
   WITH ALGORITHM = AES_256
   ENCRYPTION BY SERVER ASYMMETRIC KEY SQL_EKM_RSA_2048_Key_AW;
   ```
   Regenerate:
   ```sql
   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.
   ```sql
   ALTER DATABASE AdventureWorks
   SET ENCRYPTION ON;
   ```
9. Query the status of database encryption and its completion percentage.
   ```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 eencryptor_thumbprint = c.thumbprint
If the Query executed successfully message displays the integration between SQL server and SafeNet HSM 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.
1. Once the cluster is up and running, install the SafeNet Luna Network HSM client or HSM on Demand service client on both the nodes.

2. 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.

3. Install the Luna EKM client on both the nodes.

4. Configure the Luna EKM provider on both the nodes.

5. Open the SQL Server management studio to register the Luna EKM provider on the first node.

6. Setup the credential on the first node.

7. Create encryption keys using the Luna EKM provider on the first node.

8. Create a table and encrypt a column with the Luna EKM key on the first node.

9. Shutdown the first node.

10. Login to the second node and decrypt the data encrypted on the first node.

   If the data decrypts successfully, Extensible Key Management (EKM) using Luna EKM is operating correctly on the SQL Server cluster.
SafeNet HSM Integration with SQL Server
High Availability (Always On) Group

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
- Creating the Always On Availability Group
- Creating the Encryption Keys for Availability Group Database
- Creating Symmetric Key Encrypted by Asymmetric Key on SafeNet HSM
- Enabling Transparent Database Encryption using Asymmetric key on SafeNet HSM
- Adding the Encrypted Database to the Availability Group
- Rotating Keys for Transparent Database Encryption

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.

**NOTE:** 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.

### 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

To setup the Luna EKM provider, you must 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. 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 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 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:

   ```sql
   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 LunaEKMCre with SQL User or Login:

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

   **NOTE:** We recommend using a domain user on all SQL Server nodes.

   **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.

Creating the Always On Availability Group

You must 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

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:

   USE HSMDB;

3. 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 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
You can view symmetric keys by executing the following command from the SQL query window.

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

   SELECT * FROM [hsmdb].[sys].[symmetric_keys]

To encrypt with symmetric keys using the Luna EKM Provider
You can encrypt tables in the database using the Luna EKM Provider key stored on the SafeNet HSM.

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 with symmetric 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),
   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
Following types of asymmetric key can be created on SafeNet HSM from the SQL Server:
3 – SafeNet HSM Integration with SQL Server High Availability (Always On) Group

- 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 LunaEKMPProvider
   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 Luna EKM Provider**

You can view asymmetric keys by executing the following command from the SQL Query window.

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

**To encrypt with asymmetric keys using the Luna EKM Provider**

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

**To decrypt 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:
   ```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
ENCRYPTION 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.

```sql
OPEN SYMMETRIC KEY key1 DECRYPTION BY Asymmetric Key SQL_EKM_RSA_1024_Key;
```
4. Encrypt the data using the key1:

```sql
INSERT INTO dbo.test
VALUES (3,'some text',
EncryptByKey(KEY_GUID('Key1'), 'text to be encrypted'))
```
5. Decrypt the data using the key1:

```sql
SELECT id,name,CONVERT(varchar(MAX),
DecryptByKey(data))
FROM dbo.test where id = 3
```
6. Close the symmetric key.

```sql
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.

```sql
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.

```sql
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 secondary node using the following SQL command:

   ```
   use master;
   ALTER DATABASE tde SET HADR AVAILABILITY GROUP = AGroup;
   ```

4. 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 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 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.

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. 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 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.

**Configuring the SafeNet KSP**

You must 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"
   ```

2. Run the KSPConfig.exe (KSP configuration wizard) utility to register the SafeNet Luna KSP through a 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. Open the Available Slots drop-down and select the partition.

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

10. 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.

11. Open the Available Slots drop-down and select the partition.

12. Enter the partition password in Slot Password field.

13. 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.
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.

4. Enter a name for the Column Encryption Key in the **Name** field.
   
   Open 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.

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

2. Insert some values into the table.

   INSERT INTO dbo.Employee
   values( 101,'Emp1','ConfidentialData'),(102,'Emp2','PrivateData');

3. View the table contents in plain text.

   Select * from dbo.Employee;


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.
   Select `*` from `dbo.Employee;`
   Now the Columns data and id 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.
4. On this tab, enter the following value:
   Column Encryption Setting=Enabled

5. Click **Connect**.
6. Run the following query:
   ```sql
   use Test;
   Select * from dbo.Employee;
   ```

7. The **Parameterization for Always Encrypted** window displays. Choose your option and proceed. 
   Now unencrypted data of table displays.

   This completes the configuration of SQL Server Always Encrypted securing column master key in a SafeNet Luna HSM or HSMoD service and encrypting the column encryption key 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
"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 CREATION_DISPOSITION=OPEN_EXISTING:

```
CREATE ASYMMETRIC KEY <Name of the SQL Key>
FROM Provider LunaEKMProvider
WITH PROVIDER_KEY_NAME = '<Name of the HSM Key>',
CREATION_DISPOSITION=OPEN_EXISTING
```

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.