

# **Microsoft SQL Server**

#### nShield® HSM Integration Guide

2025-06-05

Member of Microsoft Intelligent Security Association

Microsoft Security

© 2025 Entrust Corporation. All rights reserved.

## Table of Contents

| 1. Introduction  | 1  |
|--|----|
| 1.1. Product configurations  | 1  |
| 1.2. Supported nShield hardware and software versions                | 1  |
| 1.3. Supported nShield SQLEKM provider:                              | 1  |
| 1.4. Supported nShield functionality                                 | 2  |
| 1.5. Requirements  | 2  |
| 1.6. Terms   | 3  |
| 2. Install and configure the Entrust nShield HSM                     | 4  |
| 2.1. Install the Entrust nShield HSM                                 | 4  |
| 2.2. Install the Security World software and create a Security World | 4  |
| 2.3. Install the nShield nDSOP                                       | 5  |
| 2.4. Select the protection method                                    | 5  |
| 2.5. Create the Operator Card Set (OCS) or Softcard                  | 6  |
| 3. Configure SQL EKM   | 9  |
| 3.1. Enable EKM and register the SQLEKM provider                     | 9  |
| 3.2. Verify the SQLEKM provider configuration                        | 0  |
| 3.3. Create the user SQL Server credential                           | 12 |
| 4. Configure TDE   | 15 |
| 4.1. Create a TDEKEK   | 15 |
| 4.2. Create a TDE login and credential                               | 6  |
| 4.3. Create the TDEDEK and switch on encryption                      | 9  |
| 4.4. Key rotation - Replace the TDEKEK                               | 21 |
| 4.5. Key rotation - Replace the TDEDEK                               | :4 |
| 5. Column level encryption   | 6  |
| 5.1. Create a new key  | 6  |
| 5.2. Import an existing key  | 27 |
| 5.3. Encrypt a column with a symmetric key                           | 9  |
| 5.4. Encrypt a column with an asymmetric key                         | 31 |
| 5.5. Encrypt a column with the imported asymmetric key               | 3  |
| 6. Perform backup and recovery 3                                     | 5  |
| 6.1. Back up the Security World 3                                    | 5  |
| 6.2. Restore the Security World                                      | 5  |
| 6.3. Back up the database  | 5  |
| 6.4. Restore the database  | 57 |
| 7. Upgrade nDSOP from v1.0 to v2.1.0                                 | 41 |
| 7.1. Procedure   | 41 |
| 8. Upgrade nDSOP from v2.1.0 to v2.1.1                               | 17 |

| 8.1. Procedure  |
|---|
| 9. Troubleshoot   |
| 9.1. Microsoft SQL Server, Error: 15209 while rotating the TDEKEK |
| 10. Additional resources and related products                     |
| 10.1. nShield Connect   |
| 10.2. nShield as a Service  |
| 10.3. nShield Database Option Pack                                |
| 10.4. Entrust products  |
| 10.5. nShield product documentation53                             |

## **Chapter 1. Introduction**

This document describes how to integrate Microsoft SQL Server with the nShield Database Security Option Pack (nDSOP) using an Entrust nShield hardware security module (HSM) as a Root of Trust. Entrust nShield HSMs (referred to as HSM in this guide) provide FIPS certified solutions to generate and secure the keys used to encrypt and decrypt the database.

### 1.1. Product configurations

| Product   | Version                              |
|---|--------------------------------------|
| Base OS   | Windows Server Datacenter 2025       |
| SQL Server  | Microsoft SQL Server Enterprise 2022 |
| Microsoft SQL Server<br>Management Studio<br>(SSMS) | v20.2.1                              |

Entrust tested the integration with the following versions:

# 1.2. Supported nShield hardware and software versions

Entrust has successfully tested with the following nShield hardware and software versions:

| HSM        | Security World<br>Software | Firmware                       | Netimage |
|------------|----------------------------|--------------------------------|----------|
| Connect XC | 13.6.8                     | 13.4.5 (FIPS 140-3 certified)  | 13.6.7   |
| nShield 5c | 13.6.8                     | 12.72.3 (FIPS 140-2 certified) | 13.6.7   |

Security World Software v13.6.8 is the first release supporting Window Server 2025.

### 1.3. Supported nShield SQLEKM provider:

| Product | Version |
|---------|---------|
| nDSOP   | v2.1.1  |

#### 1.4. Supported nShield functionality

| Functionality    | Support |
|------------------|---------|
| FIPS 140 Level 3 | Yes     |
| Key Management   | Yes     |
| Key Generation   | Yes     |
| Key Recovery     | Yes     |
| 1 of N Card Set  | Yes     |
| Softcards        | Yes     |
| Module Only Key  | No      |
| Fail Over        | Yes     |
| Load Balancing   | Yes     |
| nSaaS            | Yes     |

#### 1.5. Requirements

• Access to the Entrust TrustedCare Portal.

Be familiar with:

- The Microsoft SQL Server features and documentation.
- The Microsoft SQL Server Management Studio features and documentation.
- The T-SQL language. The minimum requirement for T-SQL is a basic understanding of SQL tasks such as creating a database or tables.
- Database security concepts and practices.
- The nShield Documentation.

#### 1.6. Terms

| Acronym | Definition                           |
|---------|--------------------------------------|
| SQLEKM  | SQL Server Extensible Key Management |
| TDEKEK  | TDE Key Encryption Key               |
| TDEDEK  | TDE Database Encryption Key          |

# Chapter 2. Install and configure the Entrust nShield HSM

Prerequisites:

- A Windows Server with Microsoft SQL server.
- SQL Server Management Studio installed.
- The database TestDatabase has been created and is available for the integration.

#### 2.1. Install the Entrust nShield HSM

Install the nShield Connect HSM locally, remotely, or remotely via the serial console. Condensed instructions are available in the following Entrust nShield Support articles.

- How To: Locally Set up a new or replacement nShield Connect.
- How To: Remotely Setup a new or replacement nShield Connect.
- How To: Remotely Setup a new or replacement nShield Connect XC Serial Console Model.

For detailed instructions see the nShield v13.6.8 Hardware Install and Setup. Guides.

# 2.2. Install the Security World software and create a Security World

- 1. Install the Security World software. For detailed instructions see the nShield Security World Software v13.6.8 Installation Guide.
- Add the Security World utilities path to the system path. This path is typically C:\Program Files\nCipher\nfast\bin.
- 3. Open the firewall port 9004 for the HSM connections.
- 4. If using remote administration, open firewall port 9005 for the Entrust nShield Trusted Verification Device (TVD).
- 5. Open a command window and run the following to confirm that the HSM is **operational**:

| >enquiry |      |       |
|----------|------|-------|
| Server:  |      |       |
|          | <br> | <br>/ |

```
enquiry reply flags none
enquiry reply level Six
serial number 8FE1-B519-C5AA
mode operational
...
Module #1:
enquiry reply flags UnprivOnly
enquiry reply level Six
serial number 8FE1-B519-C5AA
mode operational
...
```

6. Create your Security World if one does not already exist or copy an existing one. Follow your organization's security policy for this. For more information see Create a new Security World.



ACS cards cannot be duplicated after the Security World is created. You may want to create extras in case of a card failure or a lost card.

7. Confirm that the Security World is "Usable\*:

```
> nfkminfo
World
generation 2
state 0x37270008 Initialised Usable ...
...
Module #1
generation 2
state 0x2 Usable
...
```

### 2.3. Install the nShield nDSOP

To install the nShield nDSOP:

- 1. Mount the nDSOP\_Windows-x.x.x.iso file.
- 2. Double-click the setup file and follow the instructions.

#### 2.4. Select the protection method

The OCS or Softcard and associated passphrase will be used to authorize access to specific keys protected by the SQLEKM provider.

 Operator Cards Set (OCS) are smartcards that are presented to the physical smartcard reader of an HSM. For more information on OCS use, properties, and K-of-N values, see Operator Card Sets (OCS). • Softcards are logical tokens (passphrases) that protect they key and authorize its use. For more information on Softcards use see Softcards.

Follow your organization's security policy to select an authorization access method.

#### 2.5. Create the Operator Card Set (OCS) or Softcard

Typically, an organization's security policies dictate the use of one or the other.

#### 2.5.1. Create the OCS

A SQL Server credential (as used for EKM) maps one protecting token to one stored passphrase. It can store information for only one token at a time. Therefore an OCS card set does need to have a quorum K of one.

Recovering from a power failure requires the OCS to be inserted in the HSM or the TVD.

- Edit file C:\ProgramData\nCipher\Key Management Data\config\cardlist to add the serial number of the card(s) to be presented or the wildcard value.
- 2. Open a command window as administrator.
- 3. Run the **createocs** utility as described below. Enter a passphrase or password at the prompt. Use the same passphrase for all the OCS cards in the set (one for each person with access privilege, plus the spares). In this example note that slot 2, remote via a TVD, is used to present the card.



After an OCS card set has been created, the cards cannot be duplicated. You may want to create extras in case of a card failure or a lost card.

Add the **-p** (persistent) option to the command below to be able to encrypt/decrypt the database after the OCS card has been removed for safe storage from either the HSM front panel slot or from the TVD. See the Preload Utility for more information.

```
> createocs -m1 -s2 -N testOCS -Q 1/1
FIPS 140-2 level 3 auth obtained.
Creating Cardset:
Module 1: 0 cards of 1 written
```

```
Module 1 slot 0: Admin Card #1
Module 1 slot 2: blank card
Module 1 slot 3: empty
Module 1 slot 2:- passphrase specified - writing card
Card writing complete.
cardset created; hkltu = edb3d45a28e5a6b22b033684ce589d9e198272c2
```

The authentication provided by the OCS as shown in the command line above is non-persistent and only available while the OCS card is inserted in the HSM front panel slot or the TVD. If the TVD loses connection to the Remote Administration client the database will be inaccessible.

4. Verify the OCS created:

```
> nfkminfo -c
Cardset list - 1 cardsets: (P)ersistent/(N)ot, (R)emoteable/(L)ocal-only
Operator logical token hash k/n timeout name
edb3d45a28e5a6b22b033684ce589d9e198272c2 1/1 none-NL testOCS
```

#### 2.5.2. Create the Softcard

A SQL Server credential (as used for EKM) maps one protecting token to one stored passphrase. Softcards are singular and do not have a quorum, so the SQL Server credential matches them quite well.

Unlike OCS protection, which requires a smart card and a passcode, a softcard does not require additional input for recovery after a power failure.

 Ensure the C:\Program Files\nCipher\nfast\cknfastrc file exists with the following content. Otherwise, create it.

```
> type "C:\Program Files\nCipher\nfast\cknfastrc"
CKNFAST_LOADSHARING=1
```

2. Execute the following command. Enter a passphrase at the prompt.

```
> ppmk -n testSC
Enter new pass phrase:
Enter new pass phrase again:
New softcard created: HKLTU 925f67e72ea3c354cae4e6797bde3753d24e7744
```

3. Verify the Softcard created:

```
> nfkminfo -s
SoftCard summary - 1 softcards:
Operator logical token hash
```

name

925f67e72ea3c354cae4e6797bde3753d24e7744 testSC

The **rocs** utility shows the OCS and Softcard created:

```
> rocs
'rocs' key recovery tool
Useful commands: 'help', 'help intro', 'quit'.
rocs> list cardset
No. Name Keys (recov) Sharing
1 testOCS 0 (0) 1 of 5
2 testSC 0 (0) (softcard)
rocs> quit
```

# Chapter 3. Configure SQL EKM

### 3.1. Enable EKM and register the SQLEKM provider

To enable EKM and register the SQLEKM provider:

- 1. Launch the SQL Server Management Studio GUI.
- 2. Enable EKM by executing the following query:

| <pre>sp_configure 'show advanced', 1 G0 RECONFIGURE G0 sp_configure 'EKM provider enabled', 1 G0 RECONFIGURE G0</pre>   |   |  |  |
|---|---|--|--|
| SQLQuery2.sql - MSSQLNSHIELD25\MSSQLI         File       Edit       View       Project       Tools       Window         • • • • •       Image: Im | NSHIELDHSM.TestDatabase (INTEROP\Administrator (66))* - Microsoft SQL Server Mana Quick Launch (Ctrl+Q) ア - ロ ><br>v Help<br>w Query 過 盈 盈 盈 盈 ② ※ む む ? ・ ? ・ 図 ・ 声 ・  |  |  |
| Object Explorer <ul> <li># X</li> <li>Connect</li> <li># X</li> <li># Connect</li> <li># Y</li> <li># Nassement</li> <li># Integration Services Catalogs</li> <li># SOL Server Agent (Agent XPs disabled)</li> <li># Y</li> <li># XEvent Profiler</li> <li># XEvent Profiler</li> </ul> <ul> <li># XEvent Profiler</li> </ul>   | SQLQuery2.sql - MSAdministrator (66))* * ×<br>sp_configure 'show advanced', 1<br>GO<br>RECONFIGURE<br>GO<br>100 % ~<br>Messages<br>Configuration option 'show advanced options' changed from 0 to 1. Run the RECONFIGURE statement to install.<br>Configuration option 'show advanced options' changed from 0 to 1. Run the RECONFIGURE statement to install.<br>Configuration option 'skew provider enabled' changed from 0 to 1. Run the RECONFIGURE statement to install.<br>Completion time: 2025-04-22T10:22:20.3275489-04:00<br>100 % ~ |  |  |

3. Register the SQLEKM provider with the SQL Server by executing the following query:

```
CREATE CRYPTOGRAPHIC PROVIDER nDSOP
FROM FILE = 'C:\Program Files\nCipher\nfast\bin\ncsqlekm.dll'
```

 Check the SQLEKM provider is listed in the SQL Server Management Studio GUI. Go to Security > Cryptographic Providers. nDSOP should be visible. Right-click it to verify that it is enabled.



#### 3.2. Verify the SQLEKM provider configuration

To verify the SQLEKM provider configuration:

1. Run the following query:

| <pre>SELECT * FROM sys.cryptographic_providers;</pre>  |         |  |  |
|--|---------|--|--|
|  |         |  |  |
| 🥐 ~vs4993.sql - MSSQLNSHIELD25\MSSQLNSHIELDHSM.master (INTEROP\Administrator (57))* - Microsoft SQL Server Manageme Quick Launch (Ctrl+Q) 🔎 🗕 🗖<br>e Edit View Query Project Tools Window Help | ×       |  |  |
| © ▼ ◎   記 ▼ 🖆 📽   இ New Query இ இ இ இ இ இ இ   光 凸 白   ク ▼ ℂ ▼   図   ▼ / ♬ 🔹 🔹   河  ᆃ ▷ ▼<br>〒 🌾   master 🔹   ▶ Execute = ✔ 訳 回 🗐 訳 訳 評   過 顧 品   雪 注   王 丞   物 ݷ                               | ÷       |  |  |
| bject Explorer   | ★ +++ < |  |  |
|  |         |  |  |

Verify the following:

- The version matched that of the nDSOP installation iso.
- Path to **dll** is correct.
- is\_enabled column set to 1.

2. Run the following query:

SELECT \* FROM sys.dm\_cryptographic\_provider\_properties;

| ~vs4993.sql - MSSQLNSHIELD25\MSSQL                                   | NSHIELDHSM.master (INTEROP\Administrator (57))* - Microsoft SQL Server Manage Quick Launch (Ctrl+Q)            |  |  |
|--|--|--|--|
| File Edit View Query Project Tools Window Help                       |  |  |  |
| 🕺 O - O   🎌 - 🖆 - 🏜 🗳   🗿  | New Query 🗿 🗟 🗟 🏫 🛣 🔐 👗 🗗 🗇 💙 - 🤍 - 🛛 🔛 🦻 🗾 - 🗐 🗐 🗐  |  |  |
| 🕴 🕆 🐂 🔤 master 🕞 🖡   | · Execute 🗉 🗸 🐯 🗊 🔡 🗊 📓 📾 🗊 🍹 🦉 🖅 洒 🖢 🖕  |  |  |
| Object Explorer 👻 👎 🗙  | ∼vs4993.sql - MSSQdministrator (57))* 😛 🗙  |  |  |
| Connect 👻 🏺 🎽 🔳 🍸 🖒 🚸  | SELECT * FROM sys.dm_cryptographic_provider_properties;  |  |  |
| ■ ■ MSSQLNSHIELD25\MSSQLNSHIELDHSM                                   | Ê Î  |  |  |
| 🗄 🛑 Databases  |  |  |  |
|  |  |  |  |
| ⊞ ■ Server Roles   |  |  |  |
| 🗄 💻 Credentials  |  |  |  |
| Cryptographic Providers Providers                                    |  |  |  |
| ⊞ <b>≡</b> Audits  | 100.%  |  |  |
| 🗄 ≡ Server Audit Specifications                                      | I Results III Messages   |  |  |
| Server Objects      Benlication                                      | friendly_name authentication_type symmetric_key_support sym sym symmetri asymmetric_key_support asymmetric_key |  |  |
| I = Always On High Availability                                      | 1 nCipher SQLEKM Provider BASIC 1 0 0 0 1 0  |  |  |
| 🗄 🗯 Management   |  |  |  |
| Integration Services Catalogs SOL Service Agent (Agent XPs disabled) |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  | Query executed successfully. MSSQLNSHIELD25\MSSQLNSHIELD INTEROP\Administrator master 00:00:00 1 rows          |  |  |

#### Verify the following:

| Column                 | Value                   |
|------------------------|-------------------------|
| friendly_name          | nCipher SQLEKM Provider |
| authentication_type    | BASIC                   |
| symmetric_key_support  | 1                       |
| asymmetric_key_support | 1                       |

3. Verify the supported cryptographic algorithms can be queried by running the following query:

```
DECLARE @ProviderId int;
SET @ProviderId = (SELECT TOP(1) provider_id FROM sys.dm_cryptographic_provider_properties
WHERE friendly_name LIKE 'nCipher SQLEKM Provider');
SELECT * FROM sys.dm_cryptographic_provider_algorithms(@ProviderId);
GO
```

🗄 📁 Databases 🗏 🛋 Security 🗄 🖷 Logins 🗄 ≡ Server Roles 🗄 📁 Credentials Cryptographic Providers nDSOP

| ~vs4993.sql - MSSQLNSHIELD  | 25\MSSQLNSHIELDHSM.m  | naster (INTEROP\Administrator (57))* - I   | Microsoft SQL Server Manage.   | Quick Launch (Ctrl+Q)                     | ₽ - □       | ×        |
|---|---|--|--|---|-------------|----------|
| File Edit View Query Proj   | ject Tools Window   | Help   |  |   |             |          |
| G = O   🏠 = 📩 = 😩 💾   | 📔 🔎 New Query 🔎   | ) 📾 📾 📾 📾   🗶 🗗 🗂   🄊  | - 🖓 - 🛛 🔄 🏓  | -   | 🗖 🌶 🏛 🗵     |          |
| 🕴 🕴 🔰 master  | - 🕨 Execute 🔳   | ✔ \$2 🗊 📑 🚼 \$2 📰   🚍 🛢  | ▦ ◧   ▣ ལ   ⊷ 관   •  | <b>*</b> =                                |             |          |
|   |   |  |  |   |             |          |
| Object Explorer   | - Ț × <mark>∼vs4993.sql -</mark>                                    | MSSQdministrator (57))* 😕 🗙  |  |   |             | -        |
| Object Explorer<br>Connect  | ← ∓ X <mark>~vs4993.sql -</mark><br>□DECLAR                         | MSSQdministrator(57))* + ×<br>RE @ProviderId int;  | iden iden sole   |   |             | <b>*</b> |
| Object Explorer<br>Connect ┆ ×┆ = ┌ ♂ ♪<br>∃ ₪ MSSQLNSHIELD25\MSSQLNSHI                             | ▼ ╀ X <mark>~vs4993.sql -</mark><br>□ DECLAR<br>□ SET @F<br>IELDHSM | MSSQdministrator(57))* ⇒ ×<br>RE @ProviderId int;<br>ProviderId = (SELECT TOP(1) p<br>friendly name LIKE 'nCipher                                  | provider_id FROM sys.dr<br>SOLEKM Provider'):                            | m_cryptographic_provider                  | _properties | ÷        |
| Object Explorer<br>Connect → ┆ ┆┆ ■ ヾ Ċ →<br>B MSSQLNSHIELD25\MSSQLNSHI<br>⊞ ■ Databases            | ▼   | MSSQdministrator(57))* + ×<br>RE @ProviderId int;<br>ProviderId = (SELECT TOP(1) p<br>friendly_name LIKE 'nCipher<br>T * FROM sys.dm_cryptographic | provider_id FROM sys.du<br>SQLEKM Provider');<br>5_provider_algorithms(( | m_cryptographic_provider<br>@ProviderId); | _properties | +<br>4   |
| Object Explorer<br>Connect → ¥ ¥ ■ ▼ C →<br>B MSSQLNSHIELD25\MSSQLNSHI<br>■ Databases<br>■ Security |   | MSSQdministrator(57))* → ×<br>RE @ProviderId int;<br>ProviderId = (SELECT TOP(1) p<br>friendly_name LIKE 'nCipher<br>[* FROM sys.dm_cryptographic  | provider_id FROM sys.du<br>SQLEKM Provider');<br>c_provider_algorithms(( | n_cryptographic_provider<br>@ProviderId); | _properties | +        |

| 🗄 💻 Audits                            | 100 9 | /o            |               |                |             |           |                       |        |          | Þ      |
|---------------------------------------|-------|---------------|---------------|----------------|-------------|-----------|-----------------------|--------|----------|--------|
| E Server Audit Specifications         | I R   | esults 🔊 Mes  | sanes         |                |             |           |                       |        |          |        |
| 🗄 💻 Server Objects                    |       | algorithm id  | algorithm tag | kev type       | kev lenath  |           |                       |        |          |        |
| 🗄 💻 Replication                       | 1     | 4             | AES 128       | SYMMETRIC KEY  | 128         |           |                       |        |          |        |
| 🗄 📁 Always On High Availability       | 2     | 5             | AES_192       | SYMMETRIC KEY  | 192         |           |                       |        |          |        |
| 🗄 📁 Management                        | 3     | 6             | AES_256       | SYMMETRIC KEY  | 256         |           |                       |        |          |        |
| 🗄 🗯 Integration Services Catalogs     | 4     | 9             | RSA_2048      | ASYMMETRIC KEY | 2048        |           |                       |        |          |        |
| SQL Server Agent (Agent XPs disabled) | 5     | 10            | RSA_3072      | ASYMMETRIC KEY | 3072        |           |                       |        |          |        |
| ⊞ I XEvent Profiler                   | 6     | 11            | RSA_4096      | ASYMMETRIC KEY | 4096        |           |                       |        |          |        |
|                                       |       |               |               |                |             |           |                       |        |          |        |
|                                       |       |               |               |                |             |           |                       |        |          |        |
|                                       | O Qu  | uery executed | successfully. | 🖻 MSSQLNS      | HIELD25\MSS | QLNSHIELD | INTEROP\Administrator | master | 00:00:00 | 6 rows |
|                                       |       |               |               |                |             |           |                       |        |          |        |

Notice each key type has its set of valid algorithms.

| Кеу Туре   | Algorithm                    |
|------------|------------------------------|
| Symmetric  | AES_128, AES_192, ASE_256    |
| Asymmetric | RSA_2048, RSA_3072, RSA_4096 |

#### 3.3. Create the user SQL Server credential

To create the user SQL Server credential:

1. Verify the OCS or Softcard created above:



- 2. Insert the OCS in the HSM slot or TVD. If using Softcard protection, no action is needed.
- 3. Navigate to Security > Credentials in SQL Server Management Studio.
- 4. Right-click **Credentials**, then select **New Credential**.

- 5. Under **New Credential**:
  - a. Enter the **Credential name**.
  - b. For **Identity**, enter the OCS card or Softcard name.
  - c. Enter the **passphrase** of the OCS card or Softcard.
  - d. Select Use Encryption Provider.
  - e. For **Provider**, select **nDSOP**.
  - f. Select **OK**.

| 📲 New Credential                        |  |                             | -  |     | $\times$ |
|---|--|-----------------------------|----|-----|----------|
| Select a page                           | 🖵 Script 🔹 😮 Help  |                             |    |     |          |
| F General                               | Credential name:<br>Identity:<br>Password:<br>Confirm password:<br>Use Encryption Provider<br>Provider | serverCredential<br>testOCS |    |     |          |
| Connection                              |  |                             |    |     |          |
| Server:<br>mssqlnshield25\MSSQLNSHIELDF |  |                             |    |     |          |
| Connection:<br>INTEROP\Administrator    |  |                             |    |     |          |
| <b>View connection properties</b>       |  |                             |    |     |          |
| Progress                                |  |                             |    |     |          |
| Ready                                   |  |                             |    |     |          |
|   |  |                             | ОК | Can | cel      |

6. Verify the new credential in **Security > Credentials**. You may need to rightclick and select **Refresh**.

| Reference of the second | LNSHIELDHSM.master (INTEROP\Administrator (57))* - Microsoft SQL Server Manage Quick Launch (Ctrl+Q) 🔎 🗕 🗖  | x              |
|--|---|----------------|
| ◎ • ◎   数 • 1 • 🖕 💾 🔐 🍠  | New Query 🛢 📾 📾 🎰 🐇 🗗 🏦 🗇 - 🤍 - 🖾 🔤 - 🛛 🗾 - 🗐 🗾 - 🗐 🗲 🛎 🖸 -   | , <sup>±</sup> |
| 🕴 🛱 🦞 🛛 master 🔹 💽   | > Execute ■ ✔ 많 @ 읍 맘 많 @ 읍 霝 읎 [ 第 注   还 좌   ゐ ᇴ   |                |
| Object Explorer 🔹 👎 🗙  | ~vs4993.sql - MSSQdministrator (57))* + ×   | -              |
| Connect * * * * * * * * * * * * * * * * * * *  | <pre>DECLARE @ProviderId int;<br/>SET @ProviderId = (SELECT TOP(1) provider_id FROM sys.dm_cryptographic_provider_properties<br/>WHERE friendly_name LIKE 'nCipher SQLEKM Provider');<br/>SELECT * FROM sys.dm_cryptographic_provider_algorithms(@ProviderId);<br/>GO</pre> | 4⊢             |
| 🗄 🖷 Audits   |   |                |
| 🗄 💻 Server Audit Specifications  | III Results gli Messages  |                |
| 🗄 💻 Server Objects   | aigorithm_id aigorithm_idag key_type key_tength   |                |
| H = Replication  | 2 5 AES 192 SYMMETRICKET 192  |                |
| 🗄 💻 Always On High Availability  | 3 6 AES_256 SYMMETRIC KEY 256   |                |
| 🗄 💻 Management   | 4 9 RSA_2048 ASYMMETRIC KEY 2048  |                |
| Integration Services Catalogs  | 5 10 RSA_3072 ASYMMETRIC KEY 3072   |                |
| SQL Server Agent (Agent XPs disabled)  | 6 11 RSA_4096 ASYMMETRIC KEY 4096   |                |
| ⊞ 🗉 XEvent Profiler  |   |                |
|  | 🛛 Query executed successfully. 👘 MSSQLNSHIELD25\MSSQLNSHIELD INTEROP\Administrator master 00:00:00 6 rows   | ;              |

- Navigate to Security > Logins. Right-click the login used to access the TestDatabase and select Properties.
- 8. Check **Map to Credentials** in the dialog. Select the server credential created above in the drop-down to the right. Then select **Add**, and select **OK**.

| Login Properties - INTEROP  | \Administrator   |                                  | -                 |        | × |
|---|--|----------------------------------|-------------------|--------|---|
| Select a page   | 🖵 Script 🕞 😯 Help  |                                  |                   |        |   |
| <ul> <li>✔ General</li> <li>✔ Server Roles</li> <li>✔ User Mapping</li> <li>✔ Securables</li> <li>✔ Status</li> </ul> | Login name:<br>Windows authentication<br>Microsoft Entra ID authentic<br>SQL Server authentication<br>Password:          | INTEROP/Administrator            |                   | Search |   |
| Connection  | Comming password.  Specify old password Old password: Enforce password expir User must change pass Mapped to certificate | y<br>ation<br>word at next login | ~                 |        |   |
| Server:<br>mssqlnshield25\MSSQLNSHIELDI   | Mapped to asymmetric key Map to Credential   |                                  | ~                 |        |   |
| Connection:<br>INTEROP\Administrator<br>Wiew connection properties  | Mapped Credentials   | Credential<br>serverCredential   | Provider<br>nDSOP | Add    |   |
| Progress  |  |                                  |                   |        |   |
| Ready   |  |                                  |                   | Remove |   |
|   | Default database:  | master                           | ~                 |        |   |
|   |  |                                  | ОК                | Cancel |   |

## Chapter 4. Configure TDE

The TDE Database Encryption Key (TDEDEK) is a symmetric key that is used to perform the actual encryption of the database and are unique to a given database. It is created by SQL Server and cannot be exported from the database, meaning it cannot be created or directly protected by the SQLEKM provider (nShield HSM).

The TDEDEK is protected within the database by encrypting it with a wrapping key. The wrapping key is called the TDE Key Encryption Key (TDEKEK). The TDEKEK is an asymmetric key protected by the SQLEKM provider in the nShield HSM. It is possible to have a single TDEKEK for multiple databases, or different TDEKEKs for different databases.

The TDEKEK must be created under the tdeLogin/tdeCredential. However, the current user does not have to use the tdeCredential, so long as the user credential is using the same OCS or Softcard as the tdeCredential.

#### 4.1. Create a TDEKEK

To create a TDEKEK in the master database:

- 1. Insert the OCS in the HSM slot or TVD. If using Softcard protection, no action is needed.
- 2. Run the following query:

```
USE master;
CREATE ASYMMETRIC KEY "<name_of_key_in_database>"
FROM PROVIDER "<SQLEKM_provider>"
WITH
PROVIDER_KEY_NAME = '<name_of_key_in_SQLEKM_provider>',
CREATION_DISPOSITION = CREATE_NEW,
ALGORITHM = <asymmetric_algorithm_desc>;
GO
```

Where:

| name_of_key_in_database            | Name given to the key in the database.        |
|------------------------------------|---|
| name_of_key_in_SQLEKM_provid<br>er | Name given to the key in the SQLEKM provider. |
| asymmetric_algorithm_desc          | A valid asymmetric key algorithm descriptor.  |

For example:

| USE master;   |
|---|
| CREATE ASYMMETRIC KEY "AsymTestWrappingKeyDatabase"         |
| FROM PROVIDER "nDSOP"                                       |
| WITH  |
| <pre>PROVIDER_KEY_NAME = 'AsymTestWrappingKeySQLEKM',</pre> |
| CREATION_DISPOSITION = CREATE_NEW,                          |
| ALGORITHM = RSA_2048;                                       |
| GO  |

Notice the newly created key highlighted in the object explorer.



3. The key generated can also be verified using a CLI command:



The rocs utility shows the names and protection methods of the keys.

```
>rocs
`rocs' key recovery tool
Useful commands: `help', `help intro', `quit'.
rocs> list keys
No. Name App Protected by
1 AsymTestWrappingKeySQLEK simple testOCS
rocs> exit
```

#### 4.2. Create a TDE login and credential

A tdeLogin and tdeCredential allows an ordinary database user, who is fully authorized to use the database, but has no SQLEKM credentials of their own, to perform query operations using a TDE encrypted database. Without the tdeLogin and tdeCredential, then every user would need their own credentials. It is beyond the scope of this document to provide an example of how to use these credentials, only on how to create them.

#### 4.2.1. Create a TDE credential

To create a TDE credential:

- 1. Insert the OCS in the HSM slot or TVD. If using Softcard protection, no action is needed.
- 2. In SQL Server Management Studio, navigate to **Security > Credentials**.
- 3. Right-click **Credentials**, then select **New Credential**.
- 4. Under New Credential:
  - a. Enter the **Credential name**.
  - b. For **Identity**, enter the OCS card or Softcard name.
  - c. Enter the **passphrase** of the OCS card or Softcard.
  - d. Select Use Encryption Provider.
  - e. For **Provider**, select **nDSOP**.
  - f. Select **OK**.

| 🔒 New Credential                        |                           |               | -  |    | ×     |
|---|---------------------------|---------------|----|----|-------|
| Select a page                           | 🖵 Script 🕞 😯 Help         |               |    |    |       |
| ✤ General                               |                           |               |    |    |       |
|   |                           |               |    |    |       |
|   | Credential name:          | tdeCredential |    |    |       |
|   | Identity:                 | testOCS       |    |    |       |
|   | Password:                 | *****         |    |    |       |
|   | Confirm password          | *****         |    |    |       |
|   | oomini passiona.          |               |    |    | _     |
|   | 🔽 Use Encryption Provider |               |    |    |       |
|   | Provider                  | DSOP          |    |    |       |
|   | TIONGE                    | 1000          |    |    |       |
|   |                           |               |    |    |       |
| Connection                              |                           |               |    |    |       |
| Server:<br>mssqlnshield25\MSSQLNSHIELDF |                           |               |    |    |       |
| Connection:<br>INTEROP\Administrator    |                           |               |    |    |       |
| View connection properties              |                           |               |    |    |       |
|   |                           |               |    |    |       |
|   |                           |               |    |    |       |
| Progress                                |                           |               |    |    |       |
| Ready                                   |                           |               |    |    |       |
| 7455°                                   |                           |               |    |    |       |
|   |                           |               |    |    |       |
|   |                           |               | ОК | Ca | incel |

5. Notice the credential created.



#### 4.2.2. Create a TDE login

To create a TDE login:

- 1. In SQL Server Management Studio, navigate to **Security > Logins**.
- 2. Right-click Logins, then select New Login.
- 3. Enter the **Login name**.
- 4. Select **Mapped to asymmetric key**. Then select the TDEDEK created in Create a TDEKEK.
- 5. Select **Map to Credential**. Then select the TDE credential created in Create a TDE credential. Then select **Add**.
- 6. Select **OK**.

| Select a page                | 🖵 Script 👻 😯 Help                            |                       |           |        |
|------------------------------|--|-----------------------|-----------|--------|
| General                      |  |                       |           |        |
| Server Roles                 |  |                       |           |        |
| Securables                   | Login name:                                  | taeLogin              |           | Search |
| ✤ Status                     | Windows authentication                       |                       |           |        |
|                              | Microsoft Entra ID authentic                 | cation                |           |        |
|                              | SQL Server authentication                    |                       |           |        |
|                              | Password:                                    |                       |           |        |
|                              | Confirm password                             |                       |           |        |
|                              | Specify old password                         |                       |           |        |
|                              |  |                       |           |        |
|                              | Old password:                                |                       |           |        |
|                              | Enforce password policy                      | /                     |           |        |
|                              | Enforce password expira                      | ation                 |           |        |
|                              | User must change pass                        | word at next login    |           |        |
| Connection                   | <ul> <li>Mapped to certificate</li> </ul>    |                       | $\sim$    |        |
| Server:                      | <ul> <li>Mapped to asymmetric key</li> </ul> | AsymTestWrappingKeyDa | atabase ~ |        |
| mssqlnshield25\MSSQLNSHIELDI | Map to Credential                            | serverCredential      | ~         |        |
| Connection:                  |  |                       |           | Add    |
| INTEROP\Administrator        | Mapped Credentials                           | Credential            | Provider  |        |
| View connection properties   |  | tdeCredential         | nDSOP     |        |
|                              |  |                       |           |        |
|                              |  |                       |           |        |
|                              |  |                       |           |        |
| Progress                     |  |                       |           |        |
| Ready                        |  |                       |           |        |
| A <sup>4B</sup> A            |  |                       | _         | Remove |
|                              | Default database                             | master                | ~         |        |
|                              | Deraul UdldDd58.                             |                       |           |        |

7. Notice the login created.



#### 4.3. Create the TDEDEK and switch on encryption

To create the TDEDEK and switch on encryption:

1. In SQL Server Management Studio, navigate to **Databases > TestDatabase**.

- 2. Right-click TestDatabase, then select **Tasks > Manage Database Encryption**.
- 3. Set Encryption Algorithm to AES 256 or your choice.
- 4. Select **Use server asymmetric key**. Then select the TDEDEK created in Create a TDEKEK.
- 5. Select **Set Database Encryption On**. If using OCS protection, present the card at this point. Then select **OK**.

| 🔉 Manage Database Encryption                             |                               |             | -         |         | ×      |
|--|-------------------------------|-------------|-----------|---------|--------|
| 🕕 Ready  |                               |             |           |         |        |
| Select a page  | 🗊 Script 🝷 😯 Help             |             |           |         |        |
| 🔑 General  | Database Encryption Key       |             |           |         |        |
|  | Encryption Algorithm:         | AES 256     |           |         | $\sim$ |
|  | O Use server certificate:     |             |           |         | $\sim$ |
|  | O Use server asymmetric key:  | AsymTestWra | ppingKeyD | atabase | ~      |
|  | Database Encryption Options — |             |           |         |        |
| Connection   |                               |             |           |         |        |
| HSQLNSHIELD25MSSQLNSHI<br>ELDHSM [INTEROP/Administrator] | Set Database Encryption Or    | ı           |           |         |        |
| View connection properties                               |                               |             |           |         |        |
| Progress   |                               |             |           |         |        |
| Ready  |                               |             |           |         |        |
|  | C                             | ок с        | ancel     | Help    | >      |

6. Run the following query to verify the encryption state:

| /****** Script <b>for</b> SelectTopNRows command from SSMS  |
|---|
| 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.systement to a construct on element $p_{10}$ induced in the contraction of the construction of the contraction of the contractio |

| SQLQuery1.sql - MSSQLNSHIELD25\MSSQLNSHIELD | HSM.TestDatabase (INTEROP\Administrator (57))* - Microsoft SQL Server Quick Launch (Ctrl+Q)   | - 🗆 ×               |
|---|---|---------------------|
| File Edit View Query Project Tools Windo    | ow Help   |                     |
| 🖉 🗢 🗢 📔 👻 🎦 - 😩 🔛 🔐 🖡 New Query             | 🛢 励 励 励 🕆 ひ 白   🤊 - 🦿 🔗 🕞 🥬   | · 🖻 🖂 - 📮           |
| 🕴 🙀 TestDatabase 🔹 🕨 Execute                | = ✔ 器 酉 🔒 器 酽 📾  邱 🧵 🦉 포 좌 🐌 –  |                     |
| Object Explorer 🔹 👎 🗙                       | SQLQuery1.sql - MSAdministrator (57))* + ×  | -                   |
| Connect X # X# = X C                        | /****** Script for SelectTopNRows command from SSMS ******/   | ÷                   |
|   | SELECT DB_NAME(e.database_id) AS DatabaseName, e.database_id, e.encryption_st   | ate, CASI 🔺         |
| MSSQLNSHIELD25\MSSQLNSHIELDHSM (SQL Server  | WHEN 0 THEN 'No database encryption key present, no encryption'   |                     |
| Databases                                   | WHEN 1 THEN 'Unencrypted'   |                     |
| System Databases                            | WHEN 2 THEN Encryption in progress  |                     |
|   | WHEN 4 THEN 'Key change in progress'  |                     |
|   | WHEN 5 THEN 'Decryption in progress'  |                     |
| E Server Objects                            | END AS encryption_state_desc, c.name, e.percent_complete FROM sys.dm_database   | <pre>encrypt:</pre> |
|   | LEFT JOIN master.sys.certificates AS c ON e.encryptor_thumbprint = c.thumbpri   | int                 |
| Always On High Availability                 |   |                     |
| III = Management                            |   | P                   |
| Integration Services Catalogs               | 🖩 Results 📓 Messages  |                     |
| 县 SQL Server Agent (Agent XPs disabled)     | DatabaseName         database_id         encryption_state         encryption_state_desc         name         percent_complete           1         tempdb         2         3         Encryption_state_desc         NULL         0 |                     |
| ⊞ ≝ XEvent Profiler                         | 2 TestDatabase 5 3 Encrypted NULL 0   |                     |
|   |   |                     |
|   |   |                     |
|   |   |                     |
|   |   |                     |
|   |   |                     |
|   |   |                     |
|   | 🛿 Query execute 🍙 MSSQLNSHIELD25\MSSQLNSHIELD INTEROP\Administrator TestDatabase 00:00  | 1:00 2 rows         |

The following table shows the value returned for encryption state and the meaning.

| Encryption state | Meaning   |
|------------------|---|
| 0                | Encryption disabled (or no encryption key)  |
| 1                | Unencrypted or Decrypted  |
| 2                | Unencrypted or Decrypted  |
| 3                | Encrypted   |
| 4                | Key change in progress  |
| 5                | Decryption in progress  |
| 6                | Protection change in progress (The certificate or<br>asymmetric key that is encrypting the database<br>encryption key is being changed) |

7. Turn encryption off and on by toggling **Set Database Encryption On** in the steps above, each time running the query to verify the encryption state.

#### 4.4. Key rotation - Replace the TDEKEK

This is the wrapping key called TDE Key Encryption Key, an asymmetric key protected by the SQLEKM provider in the nShield HSM.

1. Create a new asymmetric TDEKEK. Follow the procedure in Create a TDEKEK. For example:

```
USE master;
CREATE ASYMMETRIC KEY "AsymTestWrappingKeyDatabase2"
FROM PROVIDER "nDSOP"
WITH
PROVIDER_KEY_NAME = 'AsymTestWrappingKeySQLEKM2',
CREATION_DISPOSITION = CREATE_NEW,
ALGORITHM = RSA_2048;
GO
```

```
> nfkminfo -l
```

```
Keys protected by cardsets:
key_simple_sqlekm-edb3d45a28e5a6b22b033684ce589d9e198272c2-2ee1a00e203e99bad707a15766f823fb7b6df6c5
`AsymTestWrappingKeySQLEKM2'
key_simple_sqlekm-edb3d45a28e5a6b22b033684ce589d9e198272c2-6e2dc27a8a41f9159d157986e157df87b9bbfcc7
`AsymTestWrappingKeySQLEKM'
```



- 2. Create a new TDE credential. Follow the procedure in Create a TDE credential.
- 3. Create a new TDE login mapped to the new asymmetric key and credential created above. Follow the procedure in Create a TDE login.

For example:

| Login - New   |   |  | —                 |        |
|---|---|--|-------------------|--------|
| Select a page   | 🖵 Script 🕞 😯 Help   |  |                   |        |
| <ul> <li>✗ General</li> <li>✗ Server Roles</li> <li>✗ User Mapping</li> <li>✗ Securables</li> <li>✗ Status</li> </ul> | Login name:<br>Vindows authentication<br>Microsoft Entra ID authentic<br>SQL Server authentication<br>Password:<br>Confirm password:<br>Specify old password<br>Old password:<br>Enforce password policy<br>Enforce password expirat<br>User must change pass | tdeLogin2                                |                   | Search |
| Connection  | Mapped to certificate   |  | ~                 |        |
| Server:<br>mssqlnshield25\MSSQLNSHIELDI<br>Connection:  | <ul> <li>Mapped to asymmetric key</li> <li>Map to Credential</li> </ul>   | AsymTestWrappingKeyD<br>serverCredential | vatabase2 v       | Add    |
| INTEROP/Administrator   | Mapped Credentials  | Credential<br>tdeCredential2             | Provider<br>nDSOP |        |
| Progress<br>Ready   |   |  |                   |        |
|   | Default database:   | master                                   | ~                 | Remove |
|   |   |  | ОК                | Cancel |

- 4. In SQL Server Management Studio, navigate to **Databases > TestDatabase**.
- 5. Right-click TestDatabase, then select **Tasks > Manage Database Encryption**.
- 6. Select **Re-Encrypt Database Encryption Key** and **Use server asymmetric**.
- 7. Select the newly created asymmetric key **AsymTestWrappingKeyDatabase2**.
- 8. Deselect Regenerate Database Encryption Key.
- 9. Select Set Database Encryption On.
- 10. Select **OK**.

| 尽 Manage Database Encryption                              |                                | -                            |   | ×      |
|---|--------------------------------|------------------------------|---|--------|
| 🕕 Ready   |                                |                              |   |        |
| Select a page   | 🖵 Script 🝷 😮 Help              |                              |   |        |
| <ul> <li>General</li> <li>Properties</li> </ul>           | Encryption Key Option          |                              |   |        |
|   | < Re-Encrypt Database Encrypti | on Key:                      |   |        |
|   | Use server certificate:        |                              |   | $\sim$ |
|   | O Use server asymmetric key:   | AsymTestWrappingKeyDatabase2 |   | $\sim$ |
|   | Regenerate Database Encrypt    | ion Key:                     |   |        |
|   | Encryption Algorithm:          | AES 256                      |   | $\sim$ |
|   | Database Encountion Ontion     |                              |   |        |
| Connection  |                                |                              |   |        |
| SSQLNSHIELD25/MSSQLNSHI<br>ELDHSM [INTEROP/Administrator] | Set Database Encryption On     |                              |   |        |
| View connection properties                                |                                |                              |   |        |
| Progress  |                                |                              |   |        |
| Ready   |                                |                              |   |        |
|   |                                | OK Cancel                    | ŀ | lelp   |

11. Verify the encryption state as shown in Create the TDEDEK and switch on encryption.

#### 4.5. Key rotation - Replace the TDEDEK

This is the key called TDE Database Encryption Key, a symmetric key used to perform the actual encryption of the database. It is created by SQL Server and cannot be exported from the database. It is protected within the database by encrypting it with a wrapping key TDEKEK.

- 1. In SQL Server Management Studio, navigate to **Databases > TestDatabase**.
- 2. Right-click TestDatabase, then select **Tasks > Manage Database Encryption**.
- 3. Deselect Re-Encrypt Database Encryption Key.
- 4. Select Regenerate Database Encryption Key.
- 5. Select **AES 256**.
- 6. Select Set Database Encryption On.
- 7. Select **OK**.

| 尽 Manage Database Encryption                                 |   |                         | -      |     | × |
|--|---|-------------------------|--------|-----|---|
| 🕕 Ready  |   |                         |        |     |   |
| Select a page  | 🗊 Script 🔹 😮 Help                           |                         |        |     |   |
| Seneral<br>Properties  | Encryption Key Option                       |                         |        |     |   |
|  | Re-Encrypt Database Encrypt                 | ion Key:                |        |     |   |
|  | <ul> <li>Use server certificate:</li> </ul> |                         |        |     |   |
|  | Use server asymmetric key:                  | AsymTestWrappingKeyData | abase2 |     |   |
|  | Regenerate Database Encryp                  | tion Key:               |        |     |   |
|  | Encryption Algorithm:                       | AES 256                 |        |     | ~ |
|  |   |                         |        |     |   |
| Connection   | Database Encryption Option                  |                         |        |     |   |
| H MSSQLNSHIELD25/MSSQLNSHI<br>ELDHSM [INTEROP/Administrator] | Set Database Encryption On                  |                         |        |     |   |
| View connection properties                                   |   |                         |        |     |   |
| Progress   |   |                         |        |     |   |
| Ready  |   |                         |        |     |   |
|  |   | ок с                    | ancel  | Hel | р |

8. Verify the encryption state as shown in Create the TDEDEK and switch on encryption.

## Chapter 5. Column level encryption

Table column data can be protected by an Entrust nShield HSM protected key. These nDSOP EKM keys can encrypt/decrypt data in a column.

#### 5.1. Create a new key

Create a new key within the SQL Server database to encrypt a column. This key will be protected by the Entrust nShield HSM.

- 1. Insert the OCS in the HSM slot or TVD. If using Softcard protection, no action is needed.
- 2. To create an symmetric key, run the following query:

```
USE TestDatabase;
CREATE SYMMETRIC KEY "DBSymKey"
FROM PROVIDER "nDSOP"
WITH
PROVIDER_KEY_NAME = 'EKMSymKey', IDENTITY_VALUE = '$DBSymKey',
CREATION_DISPOSITION = CREATE_NEW,
ALGORITHM = AES_256;
GO
```

3. To create a asymmetric key, run the following query:

```
USE TestDatabase;
CREATE ASYMMETRIC KEY "DBASymKey"
FROM PROVIDER "nDSOP"
WITH
PROVIDER_KEY_NAME = 'EKMASymKey',
CREATION_DISPOSITION = CREATE_NEW,
ALGORITHM = RSA_2048;
GO
```

4. Verify the keys created above.



#### > nfkminfo -l

Keys protected by cardsets: key\_simple\_sqlekm-edb3d45a28e5a6b22b033684ce589d9e198272c2-2ee1a00e203e99bad707a15766f823fb7b6df6c5 `AsymTestWrappingKeySQLEKM2' key\_simple\_sqlekm-edb3d45a28e5a6b22b033684ce589d9e198272c2-6e2dc27a8a41f9159d157986e157df87b9bbfcc7 `AsymTestWrappingKeySQLEKM' key\_simple\_sqlekm-edb3d45a28e5a6b22b033684ce589d9e198272c2-9c3e04e72445ce59c0ff3b06ee448c724c870e3d

`EKMSymKey'
key\_simple\_sqlekm-edb3d45a28e5a6b22b033684ce589d9e198272c2-a99b960c5d02a7d1505fcc3cf238634c6a990b51
`EKMASymKey'

#### 5.2. Import an existing key

The Entrust nShield HSM utility **generatekey** will be used to create an asymmetric key. Then this key will be imported in the SQL Server.

1. Run the utility **generatekey** interactive as show below:



The **ident: Key identifier? []** must begin with **sqlekm-**, and may not contain upper case characters.

```
key generation parameters:
operation Operation to perform
                                               generate
application Application
                                               simple
protect
          Protected by
                                               token
             Slot to read cards from
slot
                                               0
recovery
             Key recovery
                                               yes
verify
             Verify security of key
                                               yes
             Key type
                                               RSA
type
             Key size
                                               2048
size
             Public exponent for RSA key (hex)
pubexp
ident
             Key identifier
                                               sqlekm-ekmexistingasymkey
plainname
             Key name
                                               EKMExistingASymKey
             Blob in NVRAM (needs ACS)
nvram
                                               по
Loading cardset(s):
Module 1 slot 2: 'testOCS' #2
Module 1 slot 0: empty
Module 1 slot 3: empty
Module 1 slot 4: empty
Module 1 slot 5: empty
Module 1 slot 2:- passphrase supplied - reading card
Card reading complete.
Key successfully generated.
Path to key: C:\ProgramData\nCipher\Key Management Data\local\key_simple_sqlekm-ekmexistingasymkey
```

2. Notice the newly created key.

```
> nfkminfo -1
Keys protected by cardsets:
    key_simple_sqlekm-edb3d45a28e5a6b22b033684ce589d9e198272c2-2ee1a00e203e99bad707a15766f823fb7b6df6c5
'AsymTestWrappingKeySQLEKM2'
    key_simple_sqlekm-edb3d45a28e5a6b22b033684ce589d9e198272c2-6e2dc27a8a41f9159d157986e157df87b9bbfcc7
'AsymTestWrappingKeySQLEKM'
    key_simple_sqlekm-edb3d45a28e5a6b22b033684ce589d9e198272c2-9c3e04e72445ce59c0ff3b06ee448c724c870e3d
'EKMSymKey'
    key_simple_sqlekm-edb3d45a28e5a6b22b033684ce589d9e198272c2-a99b960c5d02a7d1505fcc3cf238634c6a990b51
'EKMASymKey'
    key_simple_sqlekm-ekmexistingasymkey 'EKMExistingASymKey'
```

3. Import the newly created key by running the following query. Re-start the SSMS if the key is not found.

```
USE TestDatabase;
GO
CREATE ASYMMETRIC KEY "DBExistingASymKey"
FROM PROVIDER "nDSOP"
WITH
PROVIDER_KEY_NAME = 'EKMExistingASymKey',
CREATION_DISPOSITION = OPEN_EXISTING;
GO
```



#### 5.3. Encrypt a column with a symmetric key

To encrypt a column with a symmetric key:

1. Consider the table TestTable in database TestDatabase.



These were created with the following script:

```
USE master
GO
-- Create database named "TestDatabase".
CREATE DATABASE TestDatabase;
GO
```

-- Create table named "TestTable" and populate it with some values. USE TestDatabase G0 CREATE TABLE TestTable (FirstName varchar(50), LastName varchar(50), Email varchar(320), Password nvarchar (50)); G0 INSERT INTO TestTable (FirstName, LastName, Email, Password) VALUES ('Firstname1', 'Lastname1', 'Firstname1.Lastname1@testserver.com', 'Paswword1'); INSERT INTO TestTable (FirstName, LastName, Email, Password) VALUES ('Firstname2', 'Lastname2', 'Firstname2.Lastname2@testserver.com', 'Paswword2'); INSERT INTO TestTable (FirstName, LastName, Email, Password) VALUES ('Firstname3', 'Lastname3', 'Firstname3.Lastname3@testserver.com', 'Paswword3'); INSERT INTO TestTable (FirstName, LastName, Email, Password) VALUES ('Firstname4', 'Lastname4', 'Firstname4.Lastname4@testserver.com', 'Paswword4'); INSERT INTO TestTable (FirstName, LastName, Email, Password) VALUES ('Firstname5', 'Lastname5', 'Firstname5.Lastname5@testserver.com', 'Paswword5'); 60 --Grants permissions to <DOMAIN>/Administrator. USE TestDatabase; GRANT ALTER ANY COLUMN ENCRYPTION KEY TO "INTEROP\Administrator"; GRANT ALTER ANY COLUMN MASTER KEY TO "INTEROP\Administrator"; GRANT VIEW ANY COLUMN ENCRYPTION KEY DEFINITION TO "INTEROP\Administrator"; GRANT VIEW ANY COLUMN MASTER KEY DEFINITION TO "INTEROP\Administrator"; 60

 Run the following query to create a new Encrypted\_Password column containing the encrypted passwords with the symmetric key created above, and populate the Password column with blanks.

```
USE TestDatabase;
ALTER TABLE TestTable
ADD Encrypted_Password VARBINARY (256);
GO
UPDATE TestTable
SET Encrypted_Password = ENCRYPTBYKEY(KEY_GUID('DBSymKey'), Password);
UPDATE TestTable
SET Password = '';
GO
```

 Notice the new Encrypted\_Password column containing the encrypted passwords.

| SQLQuery4.sql - MSSQLNSHIELD25 | MSSQLNSHIELDHSM.TestDatabase (INTEROP\Administrator (58)) - Microsoft SQL Quick Launch (Ctrl+Q) 👂 🗕 🗖 🗙  |
|--------------------------------|--|
| File Edit View Query Project   | Tools Window Help  |
| 8 • 0 者 • 🔁 • 🏜 🗳              | 』 New Query 』 읎 읎 읎 없 ㅎ 나 다 다 ! ㅋ · · · · · · · · · · · · · · · · · ·  |
| 🕴 🕆 🔰 🛛 TestDatabase 🗸 🗸       | ▶ Execute 🗉 🗸 방영 🗊 🔡 방업 방말   교실 🛲 🖓   🥶 🥶 🗮 🗃 🕮 🖓 -  |
| Object Explorer 🛛 👻 🖡 🗙        | SQLQuery4.sql - MSAdministrator (58)) 😕 🗙 SQLQuery3.sql - MSAdministrator (53))* 🗧   |
| Connect - 🛱 🎽 🔳 🍸 🖒 🚸          | SELECT TOP (1000) [FirstName]  |
| 🖃 💻 Databases                  | ,[LastName]  |
| 🗄 📁 System Databases           | , [Password]   |
| 🗄 📁 Database Snapshots         | ,[Encrypted_Password]  |
| 🗆 🖶 TestDatabase               | FROM [TestDatabase].[dbo].[TestTable]  |
| 🗄 💻 Database Diagrams          |  |
| 🖃 📁 Tables                     |  |
| 🗄 💻 System Tables              |  |
| 🗄 💻 FileTables                 |  |
| 🗄 💻 External Tables            | 100 % -  |
| 🗄 🖷 Graph Tables               | III Results 🗊 Messages   |
| 🗆 🎟 dbo.TestTable              | FirstName LastName Email Password Encrypted_Password   |
| 🗄 💻 Columns                    | 1 Firstname1 Lastname1 Firstname1.Lastname1@testserver.com 0x00CC8F338395D18B59BA817B7E3898CF020000006A4531  |
| 🗄 📫 Keys                       | 2 Firstname2 Lastname2 Firstname2.Lastname2@testserver.com 0x00CC8F338395D18B59BA817B7E3898CF020000016D37A   |
| 🗄 💻 Constraints                | 3 Firstname3 Lastname3 Firstname3.Lastname3@testserver.com 0x00CC8F338395D18B59BA817B7E3898CF02000000C3FA50  |
| 🗄 💻 Triggers                   | 4 Firstname4 Lastname4 Lastname4 Ziestserver.com 0x00CC2F338395D18E59BA817B723090C000001AB6CL  |
| 🗄 💻 Indexes                    | 5 riisunames Lasunames Lasunames Lasunames dasunames de lasunames dasunames dasu |
| 🗄 💻 Statistics                 |  |
| 🗄 🗯 Dropped Ledger Tables      |  |
| 🗄 🛋 Views                      |  |
|                                | Cuery executed successt CMSSQLNSHIELD25\MSSQLNSHIELD INTEROP\Administrator festDatabase 00:00:00 5 rows  |

4. Run the following query to decrypted the column above.

```
USE TestDatabase;
UPDATE TestTable
SET Password = DecryptByKey(Encrypted_Password);
GO
```

5. Notice the **Password** column is now populated with the decrypted password.



#### 5.4. Encrypt a column with an asymmetric key

To encrypt a column with an asymmetric key:

1. Run the following query to encrypt the passwords with the asymmetric key created above, and populate the **Password** column with blanks.



2. Notice the **Encrypted\_Password** column has new values corresponding to the asymmetric key.



3. Run the following query to decrypted the column above.



4. Notice the **Password** column is now populated with the decrypted password.

| SQLQuery7.sql - MSSQLNSHIELD25\MSSQLNSHIELDF | SM.TestDatabase (INTEROP\Administrator (72)) - Microsoft SQL Server Quick Launch (Ctrl+Q)                   |
|--|---|
| File Edit View Query Project Tools Window    | v Help  |
| 🍈 🗢 🗢 🖹 🝷 🏪 🖕 📮 New Query                    | 🛢 🗠 🗠 🔝 🕹 분리 🖞 - 오 - 🔯 🚽 🏓 🗾 - 🗍 🕶 🖸 - 🍃  |
| 🗧 🐨 TestDatabase 🔹 🕨 Execute                 |   |
|  |   |
| Object Explorer 👻 🖣 🗙                        | SQLQuery7.sql - MSAdministrator (72)) 😕 🗙 SQLQuery6.sql - MSAdministrator (54))*                            |
| Connect - 🕴 🎽 🔳 🝸 🖒 🤸                        | SELECT TOP (1000) [FirstName]   |
| MSSOL NSHIELD25\MSSOL NSHIELDHSM (SOL Sen/   | , [LastName]  |
|  | , [[mdii]   |
| E System Databases                           | [Encrypted Password]  |
|  | FROM [TestDatabase].[dbo].[TestTable]   |
| TestDatabase                                 |   |
| 🗄 🛑 Database Diagrams                        |   |
| ⊟ <b>=</b> Tables                            |   |
| 🗄 💻 System Tables                            | -   |
| ⊞ <b>=</b> FileTables                        | 100 % -   |
| 🗄 💻 External Tables                          | III Results of Messages   |
| 🗄 💻 Graph Tables                             | FirstName LastName Email Password Encrypted Password  |
| 🗄 🎟 dbo.TestTable                            | 1 Firstname1 Lastname1 Firstname1.Lastname1@testserver.com Paswword1 0xC0E3B34B97145A58F75158C565D4896A4D8  |
| 🗄 📁 Dropped Ledger Tables                    | 2 Firstname2 Lastname2 Firstname2.Lastname2@testserver.com Paswword2 0xB1925B4C7FBE71F29645BC189E4633F3641/ |
| 🗄 🖷 Views                                    | 3 Firstname3 Lastname3 Firstname3.Lastname3@testserver.com Paswword3 0xCBE08D63AFA69D2B7598D7E1A8204A95210  |
| 🗄 📁 External Resources                       | 4 Firstname4 Lastname4 Firstname4.Lastname4@testserver.com Paswword4 0x836D78781727AF9C14B2DB02E6458A5ED80  |
| 🗄 💻 Synonyms                                 | 5 Firstname5 Lastname5 Firstname5.Lastname5@testserver.com Paswword5 0x7F752318EA0540E8E2E4918CCD585B25E2A  |
| 🗄 📁 Programmability                          |   |
| 🗄 🗯 Query Store                              |   |
| 🗄 🗯 Service Broker                           |   |
|  | 🔮 Query execut 🛍 MSSQLNSHIELD25\MSSQLNSHIELD INTEROP\Administrator TestDatabase 00:00:00 5 rows             |

# 5.5. Encrypt a column with the imported asymmetric key

To encrypt a column with the imported asymmetric key:

1. Run the following query to encrypt the passwords with the imported asymmetric, and populate the **Password** column with blanks.

```
USE TestDatabase;
UPDATE TestTable
SET Encrypted_Password = ENCRYPTBYASYMKEY(ASYMKEY_ID('DBExistingASymKey'), Password);
UPDATE TestTable
SET Password = '';
GO
```

2. Notice the **Encrypted\_Password** column has new values corresponding to the imported key.

| SQLQuery8.sql - MSSQLNSHIELD25\MSSQLNSHIELDF | SM.TestDatabase (INTEROP\Administrator (80)) - Microsoft SQL Server Quick Launch (Ctrl+Q)           |
|--|---|
| File Edit View Query Project Tools Window    | v Help  |
| 💿 🔹 💿 🛛 🐮 👻 🎦 👻 🔛 🚰 🕌 New Query              | ୁ® ଛାଇଲାଇ   ୪୦୦   ୬-୯-  ⊠  -  ♬   |
| 🕴 👎 💜   TestDatabase 🔹   🕨 Execute           | ✔ \$2 目目 \$2 \$2 即 品 圖 ① 注 注 王王 秒 -   |
| Object Explorer - T ×                        | SQLQuery8.sql - MSAdministrator (80)) → × SQLQuery7.sql - MSAdministrator (72))* -                  |
| Connect * *                                  | SELECT TOP (1000) [FirstName]   |
|  | ,[LastName]   |
| MSSQLNSHIELD25\MSSQLNSHIELDHSM (SQL Serve    | ,[Email]  |
|  | ,[Password]   |
| 🗄 💻 System Databases                         | , [Encrypted_Password]  |
| Database Snapshots                           | FROM [lestDatabase].[db0].[lestlable]   |
| 🗏 🗎 TestDatabase                             |   |
| 🗄 🛑 Database Diagrams                        |   |
| 🗆 🖷 Tables                                   |   |
| 🗄 📁 System Tables                            | ·   |
| 🗄 💻 FileTables                               | 100 % -   |
| 🗄 💻 External Tables                          | 🖩 Results 🗊 Messages  |
| 🗄 ≡ Graph Tables                             | FirstName LastName Email Password Encrypted_Password  |
| 🗄 🎟 dbo.TestTable                            | 1 Firstname1 Lastname1 Firstname1.Lastname1@testserver.com 0x0DBEC109D0669144DBE5EA0FC20432E99283   |
| 🗄 🗯 Dropped Ledger Tables                    | 2 Firstname2 Lastname2 Firstname2.Lastname2@testserver.com 0x90660E1D243F01AA55E8EA87A20EEB26FEEC   |
| 🗄 💻 Views                                    | 3 Firstname3 Lastname3 Firstname3.Lastname3@testserver.com 0x591052A7E63690A610683439F802F0339A4532 |
| 🗄 💻 External Resources                       | 4 Firstname4 Lastname4 Firstname4.Lastname4@testserver.com 0x12632DC1D56A4814E821C9D533948A6F8C6F5  |
| 🗄 💻 Synonyms                                 | 5 Firstname5 Lastname5 Firstname5.Lastname5@testserver.com 0x257AF25256F13394466E7F177990A5E92F98E( |
| 🗄 💻 Programmability                          |   |
| 🗄 🖷 Query Store                              |   |
| 🗄 🖷 Service Broker                           |   |
|  | 🖉 Query execut 🍵 MSSQLNSHIELD25\MSSQLNSHIELD INTEROP\Administrator TestDatabase 00:00:00 5 rows     |

3. Run the following query to decrypted the column above.



4. Notice the **Password** column is now populated with the decrypted password.



## Chapter 6. Perform backup and recovery

A rigorous backup regimen is recommended to provide a means to recover both the database and associated keys used for encryption. Consult your corporate IT and security team for best practice and corporate policy requirements.

#### 6.1. Back up the Security World

The Security World data is inherently encrypted and does not require any further encryption operation to protect it. It can only be used by someone who has access to a quorum of the correct ACS cards, or the OCS card, Softcard, their passphrases, an nShield HSM and nShield Security World Software. Therefore, backup simply consists of making a copy of the Security World files and saving the copy in a safe location, as necessary to restore the keys used by the database.

- 1. Back up C\:ProgramData\nCipher\Key Management Data.
- 2. Securely store and keep a record of ACS and OCS cards associated with each Security World, preferable using the serial number on the cards.
- 3. The Softcard, used instead of OCS, resides in the Key Management Data folder. It is backed up at C\:ProgramData\nCipher\Key Management Data.
- 4. Keep a record of which database and which Security World backups correspond to each other.

#### 6.2. Restore the Security World

Restoring a Security World simply means restoring a backup copy of the Security World folder C\:ProgramData\nCipher\Key Management Data.

The ACS is required if the Security World being restored is not already loaded onto the HSM. See the *Installation Guide* and the *User Guide* for the HSM. A short version is available at How to locally set up a new or replacement nShield Connect.



Access to the Entrust nShield Support Portal is available to customers under maintenance. To request an account, contact nshield.support@entrust.com.

#### 6.3. Back up the database

To back up the database:

1. Create the backup devices by running the following query:



Notice the devices created.



2. Create the backup by running the following query:

```
-- Encrypted Backup
USE master;
GO
ALTER DATABASE TestDatabase
SET RECOVERY FULL;
GO
-- Back up the encrypted database
BACKUP DATABASE TestDatabase TO EncryptedTestDatabaseBackup;
GO
-- Back up the encrypted log
BACKUP LOG TestDatabase TO EncryptedTestDatabaseBackupLog;
GO
```

Notice the backup files created.

```
C:\Program Files\Microsoft SQL Server\MSSQL16.MSSQLSERVER\MSSQL\Backup>dir
Volume in drive C has no label.
Volume Serial Number is CC11-1791
Directory of C:\Program Files\Microsoft SQL Server\MSSQL16.MSSQLSERVER\MSSQL\Backup
08/16/2023 12:26 PM <DIR> .
08/02/2023 11:27 AM <DIR> .
08/16/2023 12:26 PM 4,743,680 TestDatabaseEncrypted.bak
08/16/2023 12:26 PM 86,528 TestDatabaseEncrypted.bak
2 File(s) 4,830,208 bytes
2 Dir(s) 6,326,734,848 bytes free
```

If the database is encrypted, the backup will also be encrypted. If the database is not encrypted, then the backup will not be encrypted. If you want to create an encrypted backup from a non-encrypted database, you will have to create the non-encrypted backup file, and then encrypt the file using an independent encryption tool.

#### 6.4. Restore the database

Restore a TDE encrypted database in a similar manner as an un-encrypted database. But for TDE encrypted database the Security World needs to be restored before restoring the encrypted database. The OCS, if used, needs to be inserted in the HSM before restoring the encrypted database. Otherwise, the restored database will appear as **(Restore Pending)**.

- 1. Install the Security World software and the nShield nDSOP if rebuilding the server. Do not create a Security World.
- 2. Restore the Security World.
- 3. Insert the OCS in the HSM front panel slot, or the TVD if using OCS protection.
- 4. Enable EKM and register the SQLEKM provider if rebuilding the server.
- 5. Create the SQL Server credential if rebuilding the server. The OCS and Softcard are in the restored Security World.
- 6. Verify the SQLEKM provider configuration if rebuilding the server.
- 7. Import the database wrapping key (TDEKEK) into the master database by running the following query. This is the TDEKEK last used to encrypt the database. This key should already exist in the restored Security World.

```
USE master;
GO
-- Import TDEKEK2
CREATE ASYMMETRIC KEY "AsymTestWrappingKeyDatabase2"
FROM PROVIDER "nDSOP"
WITH
```



- Recreate the TDE login and credential by running the following query. These are the TDE login and credential last used to encrypt the database. Notice the name of the OCS (nDSOPocs), and Softcard (nDSOPsoftcard) created earlier.
  - OCS:



| SQLQuery1.sql - MSSQLNSHIELD-1.master (INTEROP\Administrator (60))*         File       Edit       View       Project       Tools       Window       Help         Image: Solution of the state of  | - Microsoft SQL Server Management Studio (Administrat Quick Launch (Ctrl+Q) クーロ×<br>② (よひむ) ウマママ 図) マ 湾 ・ 「 ジーマー」 マーマー<br>1 容 窓 部 副 副 コ コ 注 王 つ マー   |
|---|--|
| Object Explorer <ul> <li>Image: Connect • Image: Connect •</li></ul> | SQLQuery1.sql - MSAdministrator (60))* * ×<br>USE master;<br>GO<br>tdeLogin2 and tcdCredential2<br>ECREATE LOGIN tdeLogin2 FROM ASYMMETRIC KEY AsymTestWrappingKeyDatabase2;<br>ECREATE CREDENTIAL tdeCredential2 WITH IDENTITY = 'testOCS', SECRET = 'nciph<br>FOR CRYPTOGRAPHIC PROVIDER nDSOP;<br>ALTER LOGIN tdeLogin2 ADD CREDENTIAL tdeCredential2;<br>GO<br>100 % • • • |
|   | 100 % • Cuery executed s MSSQLNSHIELD-1 (16.0 RTM) INTEROP\Administrator master 00:00:00 0 rows  |

• Softcard:

USE master; GO -- tdeLogin2 and tdeCredential2 CREATE LOGIN tdeLogin2 FROM ASYMMETRIC KEY AsymTestWrappingKeyDatabase2; CREATE CREDENTIAL tdeCredential2 WITH IDENTITY = 'testSC', SECRET = 'ncipher' FOR CRYPTOGRAPHIC PROVIDER nDSOP; ALTER LOGIN tdeLogin2 ADD CREDENTIAL tdeCredential2; GO

9. Restore the database by running the following query:

```
USE master
RESTORE DATABASE [TestDatabase] FROM DISK = 'C:\Program Files\Microsoft SQL
Server\MSSQL16.MSSQLSERVER\MSSQL\Backup\TestDatabaseEncrypted.bak'
GO
```

| 💫 SQLQuery1.sql - MSSQLNSHIELD-1.master (INTEROP\Administrator (61))*   | - Microsoft SQL Server Management Studio (Administrat 🛛 Quick Launch (Ctrl+Q) 💫 🗕 🗖 🗙  |
|---|--|
| File Edit View Query Project Tools Window Help  |  |
| 💮 😋 🗸 💿   🎌 👻 🐂 🍟 💾 🗳   🖨 New Query 🟮 🖓 🎲 🕰 ۵   | ›   ኤርስ   🍤 • ୧ •   🖾   •   🏓  |
| 🐘 🛱 🙀 🔤 master 🔹 🕨 Execute 🖉 🗸 🖧 🛱 🔚  | - 5° - E = 6   |
| Object Explorer - 👎 🗙   | SQLQuery1.sql - MSAdministrator (61))* 🗇 🗶   |
| Connect ♥ ♥ ♥ ♥ ♥ ♥ ♥ ♦<br>♥ MSSQLNSHIELD-1 (SQL Server 16.0.1000.6 - INTEROP\Administrator)<br>■ Databases<br>■ System Databases<br>■ Database Sapashots<br>■ Exervity<br>■ Server Objects | USE master<br>RESTORE DATABASE [TestDatabase] FROM DISK = 'C:\Program Files\Microsoft SQ a<br>GO   |
| ⊞ = Replication<br>⊞ = Always On High Availability  |  |
| ⊞ = Management  | 100 % - 4  |
| <ul> <li>Integration Services Catalogs</li> <li>ℜ SQL Server Agent (Agent XPs disabled)</li> <li>ℜ I XEvent Profiler</li> </ul>   | <pre>B<sup>W</sup> Messages Processed 568 pages for database 'TestDatabase', file 'TestDatabase' on file 1. Processed 1 pages for database 'TestDatabase', file 'TestDatabase_log' on file 1. RESTORE DATABASE successfully processed 569 pages in 0.218 seconds (20.362 MB/sec). Completion time: 2023-08-16T16:08:45.6902015-04:00</pre> |
|   | 100 % - 4  |
|   | Query executed s MSSQLNSHIELD-1 (16.0 RTM) INTEROP\Administrator master 00:00:00 0 rows  |
| 🗖 Ready Ln 3 Col 3  | Ch 3 INS   |

10. Return to multiple user mode by running the following script:



# Chapter 7. Upgrade nDSOP from v1.0 to v2.1.0

| From version | To version                      |
|--------------|---------------------------------|
| v1.0         | v2.1.0 (hotfix-Z166345-TAC1058) |

#### **Product configurations**

| Product                                   | Version                                       |
|---|---|
| Base OS                                   | Windows Server 2016 Datacenter                |
| SQL Server                                | Microsoft 2016 Enterprise with Service Pack 2 |
| Microsoft SQL Server<br>Management Studio | v18.8   |

#### Tested nShield hardware and software versions

| HSM        | Security World                            | Firmware                           | Netimage |
|------------|---|------------------------------------|----------|
| Connect XC | 12.60.11 with v2<br>Compatibility Package | 12.50.11 (FIPS 140-2<br>Certified) | 12.60.10 |

#### 7.1. Procedure

A database called TestDatabase has been created and encrypted and will be used in this procedure.

- 1. Back up the Security World.
- 2. Back up the database.
- 3. Run the following query to verify the encryption state.

```
/****** Script for SelectTopNRows command from SSMS ******/
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 'Encrypted'
WHEN 3 THEN 'Encrypted'
WHEN 3 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.certificates AS c ON e.encryptor\_thumbprint = c.thumbprint

4. Disable the EKM provider. Select **Security Cryptographic Providers**. Rightclick on the provider and select **Disable**.



5. Restart the SQL Server from the Windows MSSMS or services.

Close

Success

Disable cryptographic provider 'nDSOP

| Services                 |   |   |  |   |  |  | - 0  | × |
|--------------------------|---|---|--|---|--|--|--|---|
| File Action View         | Help  |   |  |   |  |  |  |   |
| ( <b>+ +        </b>     | à 🔒 🛛 🖬 🕨 🔲 🔢 🕨   |   |  |   |  |  |  |   |
| 🔍 Services (Local)       | Services (Local)  |   |  |   |  |  |  |   |
|                          | SQL Server (MSSQLSERVER)  | Name  |  | Description   | Status                                   | Startup Type   | Log On As  | ^ |
|                          | Stop the service<br>Pause the service<br>Restart the service<br>Description:<br>Provides storage, processing and<br>controlled access of data, and rapid<br>transaction processing. | Secure Socket     Security Acco     Sensor Data S     Sensor Data S     Sensor Monitt     Sensor Service     Server     Shell Hardwar     Smart Card D     Smart Card Re     SMMP Trap     Software Prot.     Special Admir | Tunneling Pr<br>unts Manager<br>ervice<br>oring Service<br>e<br>e Detection<br>evice Enumera<br>erroval Policy<br>ection<br>nistration Con | Provides su<br>The startup<br>Delivers dat<br>Monitors va<br>A service fo<br>Supports fil<br>Provides no<br>Manages ac<br>Creates soft<br>Allows the s<br>Enables the<br>Allows adm<br>Varifies actor | Running<br>Running<br>Running<br>Running | Manual<br>Automatic<br>Manual (Trig<br>Manual (Trig<br>Automatic<br>Automatic (T<br>Manual (Trig<br>Manual<br>Manual<br>Automatic (D<br>Manual<br>Manual<br>Manual | Local Service<br>Local System<br>Local System<br>Local Service<br>Local System<br>Local System<br>Local System<br>Local System<br>Local System<br>Local Service<br>Network Service<br>Local System |   |
| Stop and Start service S | Extended / Standard /<br>QL Server (MSSQLSERVER) on Local Cor   | SQL Serve<br>SQL Serve<br>SQL Serve<br>SQL Serve<br>SQL Serve<br>SQL Serve<br>SSDP Disc<br>SSDP Disc<br>State Rept<br>State Rept<br>State Rept  | Start<br>Stop<br>Pause<br>Resume<br>Restart<br>All Tasks<br>Refresh<br>Properties  | es sto<br>es so<br>es so<br>ervice<br>es th<br>ers n<br>es re<br>> hes a  | Running<br>Running<br>Running<br>Running | Manual<br>Disabled<br>Automatic<br>Automatic<br>Manual<br>Manual<br>Manual   | NT Service\MSS<br>NT Service\SQLS<br>Local Service<br>NT Service\SQLT<br>Local System<br>Local Service<br>Local System<br>Local System   | ~ |
|                          |   |   | Help   |   |  |  |  |   |

6. Wait for 60 seconds after the restart. Then check the database status. Notice **Recovery Pending** next to **TestDatabase**.



- Un-install nDSOP v1.0 EKM provider using the Windows Control Panel > Programs > Programs and Features.
- 8. Install nDSOP v2.1.0 EKM provider by mounting the .iso file and double-

clicking setup.

- 9. Insert the OCS in the HSM slot or TVD. No action is needed if you are using Softcard protection.
- 10. Retarget the keys by running the sqlekm\_retarget\_keys command:

```
C:\Users\Administrator>nfkminfo -k
Key list - 2 keys
AppName pkcs11
                             Ident uc79dfaf7c3311d22d240a7257e5e760ede89fbc70-
56ac051fb249f91e641b065dc12fec8a9fea2419
                             Ident uc79dfaf7c3311d22d240a7257e5e760ede89fbc70-
AppName pkcs11
c88b06f02bdca29f2a98b9c9352daf9191fc8afd
C:\Users\Administrator>sqlekm_retarget_keys --all
Found 2 keys to retarget
Retargetted: key_pkcs11_uc79dfaf7c3311d22d240a7257e5e760ede89fbc70-c88b06f02bdca29f2a98b9c9352daf9191fc8afd
Retargetted: key_pkcs11_uc79dfaf7c3311d22d240a7257e5e760ede89fbc70-56ac051fb249f91e641b065dc12fec8a9fea2419
C:\Users\Administrator>nfkminfo -k
Key list - 4 keys
AppName pkcs11
                             Ident uc79dfaf7c3311d22d240a7257e5e760ede89fbc70-
56ac051fb249f91e641b065dc12fec8a9fea2419
                             Ident uc79dfaf7c3311d22d240a7257e5e760ede89fbc70-
AppName pkcs11
c88b06f02bdca29f2a98b9c9352daf9191fc8afd
AppName simple
                             Ident sqlekm-79dfaf7c3311d22d240a7257e5e760ede89fbc70-
b1844c5bb4eadbdb1166dcdb64f4c5d59e4e408c
                              Ident sqlekm-79dfaf7c3311d22d240a7257e5e760ede89fbc70-
AppName simple
fa9380a3e111df122b0e02dd37c1233da89b8e16
```

11. Open the C:\ProgramData\nCipher\Key Management Data\local folder. Move all pkcs11 keys to another folder. Leave the simple keys in the current folder.



12. Set the new provider by running the following query:

```
--ChangeToNewProvider.sql
ALTER CRYPTOGRAPHIC PROVIDER nDSOP
FROM FILE = 'C:\Program Files\nCipher\nfast\bin\ncsqlekm.dll';
```

- GO
- Enable the EKM provider. Select Security > Cryptographic Providers. Rightclick the provider and select Enable.



14. Verify the new EKM provider version by running the following query. Notice the **provider\_version**.



- 15. Restart the SQL Server from the Windows MSSMS or services. Wait for 60 seconds after the restart.
- 16. Check and refresh database status. Notice the **Recovery Pending** message next to the TestDatabase goes away.
- Verify the encryption state by running the following query. Notice the encryption\_state\_desc shown as Encrypted.

/\*\*\*\*\*\* Script **for** SelectTopNRows **command** from SSMS \*\*\*\*\*\*/

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.certificates AS c ON e.encryptor\_thumbprint = c.thumbprint

| <ul> <li>PolyBase</li> <li>Always On High Availability</li> <li>Management</li> <li>Integration Services Catalogs</li> <li>SQL Server Agent (Agent XPs disabled)</li> <li>XEvent Profiler</li> </ul> | Image: Security Construction - state.sql - MS_SQL_EKM_3.master (sa (52)         File       Edit       View       Query       Project       Tools       Window         Image: Security       Image: Security       Image: Security       Image: Security       Image: Security       Image: Security         Image: Security       Image: Security       Image: Security       Image: Security       Image: Security       Image: Security         Image: Security       Image: Security       Image: Security       Image: Security       Image: Security       Image: Security         Image: Security       Image: Security       Image: Security       Image: Security       Image: Security         Image: Security       Image: Security       Image: Security       Image: Security       Image: Security         Image: Security       Image: Security       Image: Security       Image: Security       Image: Security         Image: Security       Image: Security       Image: Security       Image: Security       Image: Security         Image: Security       Image: Security       Image: Security       Image: Security       Image: Security         Image: Security       Image: Security       Image: Security       Image: Security       Image: Security         Image: Security       Image: Security       Image: Security | 2)) - Microsoft SQL Server Management Studio (Ad Quick Launch (Ctrl+Q)<br>Help<br>→ Help<br>→ → → → → → → → → → → → → → → → → → →   |
|--|---|---|
|  | <ul> <li>Management</li> <li>Integration Services Catalogs</li> <li>SQL Server Agent (Agent XPs disabled)</li> <li>XEvent Profiler</li> </ul>   | Image: The suits       Image: The state descent and the state descent |

# Chapter 8. Upgrade nDSOP from v2.1.0 to v2.1.1

| From version                    | To version |
|---------------------------------|------------|
| v2.1.0 (hotfix-Z166345-TAC1058) | v2.1.1     |

#### **Product configurations**

| Product                                   | Version                              |
|---|--------------------------------------|
| Base OS                                   | Windows Server 2022 Datacenter       |
| SQL Server                                | Microsoft SQL Server Enterprise 2022 |
| Microsoft SQL Server<br>Management Studio | v20.2.1                              |

#### Tested nShield hardware and software versions

| HSM        | Security World | Firmware                         | Netimage |
|------------|----------------|----------------------------------|----------|
| nShield 5c | 13.3.2         | 13.4.5 (FIPS 140-3<br>certified) | 13.6.7   |

#### 8.1. Procedure

A database called TestDatabase has been created and encrypted and will be used in this procedure.

- 1. Back up the Security World.
- 2. Back up the database.
- 3. Run the following query to verify the encryption state.

```
/****** Script for SelectTopNRows command from SSMS ******/
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 'Encrypted'
WHEN 3 THEN 'Encrypted'
WHEN 3 THEN 'Key change in progress'
WHEN 5 THEN 'Decryption 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.certificates AS c ON e.encryptor\_thumbprint = c.thumbprint

4. Disable the EKM provider. Select **Security Cryptographic Providers**. Rightclick on the provider and select **Disable**.



| 🛱 Disable provider                     | -       |   | ×                |
|--|---------|---|------------------|
| Success                                |         | 1 | Total<br>Success |
| Details:                               |         |   |                  |
| Action                                 | Status  |   | Messa            |
| Disable cryptographic provider 'nDSOP' | Success |   |                  |
|  |         |   |                  |
|  |         |   |                  |
|  |         |   |                  |
|  |         | ( | Close            |

5. Restart the SQL Server from the Windows MSSMS or services.

|                          |   |  |   |  |  |   |   | _ |   |
|--------------------------|---|--|---|--|--|---|---|---|---|
| Services                 |   |  |   |  |  |   |   |   | × |
| File Action View         | Help  |  |   |  |  |   |   |   |   |
| (= =) 🗖 🗐 🤇              | à 🗟 🛛 🖬 🕨 🔲 🕪 🕨   |  |   |  |  |   |   |   |   |
| 🔍 Services (Local)       | Q Services (Local)  |  |   |  |  |   |   |   |   |
|                          | SQL Server (MSSQLSERVER)  | Name   |   | Description  | Status                                   | Startup Type  | Log On As   |   | ^ |
|                          | Stop the service<br>Pause the service<br>Restart the service<br>Description:<br>Provides storage, processing and<br>controlled access of data, and rapid<br>transaction processing. |  | Tunneling Pr<br>unts Manager<br>ervice<br>ervice<br>e Detection<br>evice Enumera<br>moval Policy<br>ection<br>istration Con | Provides su     Manual       The startup     Running     Automal       Delivers dat     Manual       Monitors va     Manual       A service fo     Manual       Supports fil     Running     Automal       Provides no     Running     Automal       Manages ac     Automal       Creates soft     Manual       Allows the s     Manual       Enables the     Running       Automal     Manual |  | Manual<br>Automatic<br>Manual (Trig<br>Manual (Trig<br>Automatic<br>Automatic<br>Automatic (T<br>Manual (Trig<br>Manual<br>Automatic (D<br>Manual | Local Service<br>Local System<br>Local System |   |   |
| Stop and Start service S | Extended / Standard /<br>QL Server (MSSQLSERVER) on Local Cor   | Sol Serve<br>Sol Serve | Start<br>Stop<br>Pause<br>Resume<br>Restart<br>All Tasks<br>Refresh<br><b>Properties</b>                                    | es for<br>es so<br>es so<br>es so<br>er so<br>es for<br>es for<br>es re<br>> tes a   | Running<br>Running<br>Running<br>Running | Manual (Ing<br>Automatic<br>Manual<br>Disabled<br>Automatic<br>Automatic<br>Manual<br>Manual  | Local System<br>NT Service\MSS<br>NT Service\SQL1<br>Local Service<br>NT Service\SQL2<br>Local System<br>Local System<br>Local System   | S | ~ |
|                          |   |  | Help  |  |  |   |   |   |   |

6. Wait for 60 seconds after the restart. Then check the database status. Notice **Recovery Pending** next to **TestDatabase**.

| SQLQuery2.sql - MSSQLNSHIELD22.master (IN  | TEROP\Administrator (53))* - Microsoft SQL Server Management Studio (Admi Quick Launch (Ctrl+Q) 🔎 – 🗖 🗙   |
|--|---|
| File Edit View Project 100is Wilhow  | nep<br>Duery 過級級級(※中心) ♡ - ♡ -   図  -   湾   同プチキロ<br>aute = ✔ 23 目   29 23 m   品語 AD   目注  近天   き+  |
| Object Explorer       ▼ ⋕ ×         Connect ▼ ⋕ ×#       ▼ ♥ ♥ ★         ● MSSQLNSHIELD22 (SQL Server 16.0.1000.6 ∧         ● Databases         ● Databases         ● Database Snapshots         ● TestDatabase (Recovery Pending)         ● Logins         ● Security         ● Logins         ● Server Roles         ● Credentials | SQLQuery2.sql - MSAdministrator (53))* * X SQLQuery1.sql - MSAdministrator (69))*<br>/****** Script for SelectTopNRows command from SSMS ******/<br>SELECT DB_NAME(e.database_id) AS DatabaseName, e.database_id, e.encryption_state, CASE e.er A<br>WHEN 0 THEN 'Unencrypted'<br>WHEN 1 THEN 'Unencrypted'<br>WHEN 2 THEN 'Encryption in progress'<br>WHEN 3 THEN 'Encryption in progress'<br>WHEN 5 THEN 'Vecryption in progress'<br>WHEN 5 THEN 'Decryption in progress'<br>WHEN 5 THEN 'Decryption in progress'<br>END AS encryption_state_desc, c.name, e.percent_complete FROM sys.dm_database_encryption_ke<br>LEFT JOIN master.sys.certificates AS c ON e.encryptor_thumbprint = c.thumbprint |
| 🖃 🛑 Cryptographic Providers  | 100 % - 4   |
| 🏁 nDSOP  | III Results 🔊 Messages  |
| <ul> <li>              ■ Audits             ■ Server Audit Specifications             ■ Server Objects             ■ Replication             ■ Always On High Availability      </li> </ul>  | DatabaseName         database_id         encryption_state         description_state         name         percent_complete           1         tempdb         2         3         Encrypted         NULL         0           2         TestDatabase         5         3         Encrypted         NULL         0   |
| < · · · · · · · · · · · · · · · · · · ·  | Query executed successfully.     BMSSQLNSHIELD22 (16.0 RTM) INTEROP\Administrator master 00:00:00 2 rows  |

- Un-install nDSOP v2.1.0 EKM provider using the Windows Control Panel > Programs > Programs and Features.
- 8. Install nDSOP v2.1.1 EKM provider by mounting the .iso file and doubleclicking setup.
- 9. Insert the OCS in the HSM slot or TVD. No action is needed if you are using Softcard protection.
- 10. Set the new provider by running the following query:

11. Enable the EKM provider. Select **Security > Cryptographic Providers**. Rightclick the provider and select **Enable**.

| 🚝 Enable provider                     | -       |        | $\times$         |
|---------------------------------------|---------|--------|------------------|
| Success                               |         | 1<br>1 | Total<br>Success |
| Details:                              |         |        |                  |
| Action                                | Status  | Mes    | sage             |
| Enable cryptographic provider 'nDSOP' | Success |        |                  |
|                                       |         |        |                  |
|                                       |         |        |                  |
|                                       |         |        |                  |
|                                       |         |        |                  |
|                                       |         | CI     | ose              |

12. Verify the new EKM provider version by running the following query. Notice the **provider\_version**.

| <pre>SELECT * FROM sys.dm_cryptographic_provider_properties;</pre>  |
|---|
|   |
| 💦 SQLQuery1.sql - MSSQLNSHIELD22.master (INTEROP\Administrator (64))* - Microsoft SQL Server Management Studio (Admin. 🛛 Quick Launch (Ctrl+Q) 🖉 🗕 🗖 🗙  |
| File Edit View Query Project Tools Window Help  |
| 。   |
|   |
|   |
| Object Explorer V X SQLQuery1.sql - MSAdministrator (64))* + X  |
| Connect → # *# = ▼ ♂ → SELECT * FROM sys.dm_cryptographic_provider_properties; +  |
| B KSSQLNSHIELD22 (SQL Server 16.0.1000.6 A  |
| 🗄 🗏 Databases   |
| Security  |
| 🗄 🖷 Logins  |
| 🗄 🖷 Server Roles  |
| 🗄 🖷 Credentials   |
| Cryptographic Providers   |
| nDSOP 100 %   |
| Audits  Results  Messages   |
| Server Audit Specifications     provider_id_guid     provider_version sqlcypt_version friendly_name authentication_typ     control of the server |
| Server Diglets     1 03330 3406010C-05732000E-4309-706003720000 2.01.0001.00 1101.0000.00 Incline1 SQLEXM Provide BASIC   |
|   |
|   |
| Interruption Services Catalogs  |
| A SQL Server Agent (Agent XPs disabled)   |
| Query executed successfully.     MSSQLNSHIELD22 (16.0 RTM) INTEROP\Administrator master 00:00:00 1 rows   |

- 13. Restart the SQL Server from the Windows MSSMS or services. Wait for 60 seconds after the restart.
- 14. Check and refresh database status. Notice the **Recovery Pending** message next to the TestDatabase goes away.
- Verify the encryption state by running the following query. Notice the encryption\_state\_desc shown as Encrypted.

/\*\*\*\*\* Script **for** SelectTopNRows **command** from SSMS \*\*\*\*\*\*/

| 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.certificates AS c ON e.encryptor_thumbprint = c.thumbprint                           |



# 9.1. Microsoft SQL Server, Error: 15209 while rotating the TDEKEK

- 1. Restart the database.
- 2. Try again to rotate the TDEKEK.

# Chapter 10. Additional resources and related products

- 10.1. nShield Connect
- 10.2. nShield as a Service
- 10.3. nShield Database Option Pack
- 10.4. Entrust products
- 10.5. nShield product documentation