

Microsoft SQL Server Always Encrypted

nShield[®] HSM Integration Guide

2024-10-21

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. Role separation.	2
1.4. Multiple Windows user accounts on a single client server	3
1.5. Multiple client servers	3
1.6. Always Encrypted and TDE	3
2. Configure computers and accounts	4
2.1. Join the domain	4
2.2. Create domain accounts	4
3. Install and configure client	5
3.1. Select the protection method.	5
3.2. Install the Security World software and create a Security World.	5
3.3. Create the OCS or Softcard	8
3.4. Install and register the CNG provider	9
3.5. Install and configure SqlServer PowerShell module	12
3.6. Install the SQL Server Management Studio	13
3.7. Allow Active Directory user to remote login	13
4. Install and configure SQL server.	14
4.1. Install the SQL database engine	14
4.2. Create the SQL logins	15
5. Generate the encryption keys	16
5.1. Generate the Always Encrypted Column Master Key (CMK)	16
5.2. Generate My Column Master Key (MyCMK) and My Column Encryption Key	
(MyCEK) with SSMS	19
5.3. Generate MyCMK and MyCEK with PowerShell	25
6. Encrypt or decrypt a column with SSMS.	27
6.1. Encrypt a column	27
6.2. View an encrypted column	30
6.3. Remove column encryption	32
7. Encrypt or decrypt a column with PowerShell.	35
7.1. Encrypt a column	35
7.2. Remove column encryption	36
8. Test access to Always Encrypted keys by another user	37
9. Supported PowerShell SqlServer cmdlets	38
10. Additional resources and related products	10
10.1. nShield Connect	10

10.2. nShield as a Service	40
10.3. Entrust products	40
10.4. nShield product documentation	40

Chapter 1. Introduction

Always Encrypted is a feature in Windows SQL Server designed to protect sensitive data both at rest and in flight between a client application server and Azure or SQL Server database(s).

Data protected by Always Encrypted remains in an encrypted state until it has reached the client application server. This effectively mitigates man-in-the-middle attacks and provides assurances against unauthorized activity from rogue DBAs or admins with access to Azure or SQL server databases.

The nShield HSM secures the key used to protect the Column Master Key, stored in an encrypted state on the client application server.

1.1. Product configurations

Entrust successfully tested nShield HSM integration with Windows SQL Server and the Always Encrypted feature in the following configurations:

1.1.1. Remote server

Product	Version
SQL Server	Microsoft SQL Server 2022
Base OS	Windows Server 2022 Datacenter

1.1.2. Client

Product	Version
SQL Server GUI	Microsoft SQL Server Management Studio V18.8
Base OS	Windows 10 Enterprise

1.2. Supported nShield hardware and software versions

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

Product	Security World Software	Firmware	Netimage	OCS	Softcard	Module
Connect XC	13.4.5	12.50.11 (FIPS 140-2 certified) & 12.72.1 (FIPS 140-2 certified)	12.80.5 & 13.4.5	\checkmark	\checkmark	\checkmark
nShield 5c	13.4.5	13.2.2	13.2.2	\checkmark	\checkmark	\checkmark
nSaaS	12.80.4	12.72.1 (FIPS 140-2 certified)	12.80.5	\checkmark	\checkmark	\checkmark

1.3. Role separation

The generation of keys and the application of these keys for encryption or decryption are separate processes. The processes can be assigned to users with various access permissions, or Duty Roles. The table below shows the processes and duty roles with reference to the Security Administrator and the database Administrator.



Entrust recommends that you allow only unprivileged connections unless you are performing administrative tasks.

Process	Duty Role
Generating the Column Master Key (CMK) and Column Encryption Key (CEK)	Security Administrator
Applying the CMK and CEK in the database	Database Administrator

Four database permissions are required for Always Encrypted.

Operation	Description
ALTER ANY COLUMN MASTER KEY	Required to generate and delete a column master key

Operation	Description
ALTER ANY COLUMN ENCRYPTION KEY	Required to generate and delete a column encryption key
VIEW ANY COLUMN MASTER KEY	Required to access and read the metadata of the column master keys to manage keys or query encrypted columns
VIEW ANY COLUMN ENCRYPTION KEY	Required to access and read the metadata of the column encryption key to manage keys or query encrypted columns

1.4. Multiple Windows user accounts on a single client server

To enable multiple Windows user accounts on a single oclient server, ask Entrust Support for a Hotfix patch to allow multiple users to use the same always encrypted key.

1.5. Multiple client servers

Each client server wanting access to the content of the encrypted data with a given CEK must have:

- An HSM in the same Security World.
- A Hotfix patch to allow multiple users to use the same always encrypted key. Ask Entrust Support for this.
- A copy of the CMK key token stored on its local drive.

1.6. Always Encrypted and TDE

The same Security World can be used for Always Encrypted and TDE.

Chapter 2. Configure computers and accounts

Installation steps:

- 1. Join the domain.
- 2. Create domain accounts.

2.1. Join the domain

Windows authentication is used in this integration for added security. The Entrust nShield HSM solution for Microsoft SQL Always Encrypted enables keys that are associated with one user to be used by other users, providing secure access to a common database.

Both the client computer and the remote server computer must join the same Windows domain.

2.2. Create domain accounts

Create the following three Windows domain accounts:

- <domain>\<SQL Administrator>
- <domain>\dbuser
- <domain>\dbuser2

Chapter 3. Install and configure client

This installation must be performed on the client using the <domain_name>\Administrator account.

Installation steps:

- 1. Select the protection method
- 2. Install the Security World software and create a Security World
- 3. Create the OCS or Softcard
- 4. Install and register the CNG provider
- 5. Install and configure SqlServer PowerShell module
- 6. Install the SQL Server Management Studio
- 7. Allow Active Directory user to remote login

3.1. Select the protection method

OCS or Module protection can be used to authorize access to the keys protected by the HSM. Follow your organization's security policy to select which one.

3.2. Install the Security World software and create a Security World

- 1. Install the Security World software. For instructions, see the *Installation Guide* and the *User Guide* for the HSM.
- 2. Install Hotfix TAC-996 if multiple Windows user accounts need access to the same data. Contact nShield support to download the Hotfix. To perform the installation:
 - a. Open a command window as Administrator and uninstall the CNG:

```
C:\Users\Administrator.EXAMPLE>cnginstall32 --uninstall
nckspsw.dll removed.
ncpp.dll removed.
C:\Users\Administrator.EXAMPLE>cnginstall --uninstall
nckspsw.dll removed.
ncpp.dll removed.
```

- b. Reboot the server.
- c. Copy files as per the installation instructions in the Hotfix package:

C:\Users\Administrator.EXAMPLE>copy C:\Users\Administrator.EXAMPLE\Downloads\hotfix-Z155163-TAC996\hotfix-Z155163-TAC996\nfast\c\caping\vs2017-32\lib* "C:\Program Files\nCipher\nfast\c\caping\vs2017-32\lib\." C:\Users\Administrator.EXAMPLE\Downloads\hotfix-Z155163-TAC996\hotfix-Z155163-TAC996\nfast\c\caping\vs2017-32\lib\nckspsw.dll Overwrite C:\Program Files\nCipher\nfast\c\caping\vs2017-32\lib\.\nckspsw.dll? (Yes/No/All): All C:\Users\Administrator.EXAMPLE\Downloads\hotfix-Z155163-TAC996\hotfix-Z155163-TAC996\nfast\c\caping\vs2017-32\lib\nckspsw.lib C:\Users\Administrator.EXAMPLE\Downloads\hotfix-Z155163-TAC996\hotfix-Z155163-TAC996\nfast\c\caping\vs2017-32\lib\nckspsw.map C:\Users\Administrator.EXAMPLE\Downloads\hotfix-Z155163-TAC996\hotfix-Z155163-TAC996\nfast\c\caping\vs2017-32\lib\nckspsw.pdb C:\Users\Administrator.EXAMPLE\Downloads\hotfix-Z155163-TAC996\hotfix-Z155163-TAC996\nfast\c\caping\vs2017-32\lib\ncpp.dll C:\Users\Administrator.EXAMPLE\Downloads\hotfix-Z155163-TAC996\hotfix-Z155163-TAC996\nfast\c\caping\vs2017-32\lib\ncpp.lib C:\Users\Administrator.EXAMPLE\Downloads\hotfix-Z155163-TAC996\hotfix-Z155163-TAC996\nfast\c\caping\vs2017-32\lib\ncpp.map C:\Users\Administrator.EXAMPLE\Downloads\hotfix-Z155163-TAC996\hotfix-Z155163-TAC996\nfast\c\caping\vs2017-32\lib\ncpp.pdb 8 file(s) copied. C:\Users\Administrator.EXAMPLE>copy C:\Users\Administrator.EXAMPLE\Downloads\hotfix-Z155163-TAC996\hotfix-Z155163-TAC996\nfast\c\caping\vs2017-64\lib* "C:\Program Files\nCipher\nfast\c\caping\vs2017-64\lib\." C:\Users\Administrator.EXAMPLE\Downloads\hotfix-Z155163-TAC996\hotfix-Z155163-TAC996\nfast\c\caping\vs2017-64\lib\nckspsw.dll Overwrite C:\Program Files\nCipher\nfast\c\caping\vs2017-64\lib\.\nckspsw.dll? (Yes/No/All): All C:\Users\Administrator.EXAMPLE\Downloads\hotfix-Z155163-TAC996\hotfix-Z155163-TAC996\nfast\c\caping\vs2017-64\lib\nckspsw.lib C:\Users\Administrator.EXAMPLE\Downloads\hotfix-Z155163-TAC996\hotfix-Z155163-TAC996\nfast\c\caping\vs2017-64\lib\nckspsw.map C:\Users\Administrator.EXAMPLE\Downloads\hotfix-Z155163-TAC996\hotfix-Z155163-TAC996\nfast\c\caping\vs2017-64\lib\nckspsw.pdb C:\Users\Administrator.EXAMPLE\Downloads\hotfix-Z155163-TAC996\hotfix-Z155163-TAC996\nfast\c\caping\vs2017-64\lib\ncpp.dll C:\Users\Administrator.EXAMPLE\Downloads\hotfix-Z155163-TAC996\hotfix-Z155163-TAC996\nfast\c\caping\vs2017-64\lib\ncpp.lib C:\Users\Administrator.EXAMPLE\Downloads\hotfix-Z155163-TAC996\hotfix-Z155163-TAC996\nfast\c\caping\vs2017-64\lib\ncpp.map C:\Users\Administrator.EXAMPLE\Downloads\hotfix-Z155163-TAC996\hotfix-Z155163-TAC996\nfast\c\caping\vs2017-64\lib\ncpp.pdb 8 file(s) copied. C:\Users\Administrator.EXAMPLE>copy C:\Users\Administrator.EXAMPLE\Downloads\hotfix-Z155163-TAC996\hotfix-Z155163-TAC996\nfast\lib\versions\caping-atv.txt "C:\Program Files\nCipher\nfast\lib\versions\." Overwrite C:\Program Files\nCipher\nfast\lib\versions\.\caping-atv.txt? (Yes/No/All): All 1 file(s) copied.

d. Open a command window as Administrator and install the CNG:

```
C:\Users\Administrator.EXAMPLE>cnginstall32 --install
nckspsw.dll installed.
ncpp.dll installed.
C:\Users\Administrator.EXAMPLE>cnginstall --install
nckspsw.dll installed.
ncpp.dll installed.
```

- e. Reboot the server.
- 3. Add the Security World utilities path C:\Program Files\nCipher\nfast\bin to the Windows system path.
- 4. Open the firewall port 9004 for the HSM connections.
- 5. Install the nShield Connect HSM locally, remotely, or remotely via the serial console. See the following nShield Support articles and the *Installation Guide* for the HSM:
 - How to locally set up a new or replacement nShield Connect
 - How to remotely set up a new or replacement nShield Connect
 - How to remotely set up a new or replacement nShield Connect XC Serial Console model



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

6. Open a command window and run the following to confirm that the HSM is **operational**:

```
C:\Users\Administrator.EXAMPLE>enquiry
Server:
enquiry reply flags none
enquiry reply level Six
serial number 5F08-02E0-D947 6A74-1261-7843
                   operational
12.80.4
mode
version
. . .
Module #1:
enquiry reply flags none
enquiry reply level Six
serial number 5F08-02E0-D947
mode
                    operational
                    12.72.1
version
 . . .
```

- 7. Create your Security World if one does not already exist, or copy an existing one. Follow your organization's security policy for this.
- 8. Confirm that the Security World is **usable**:

```
C:\Users\Administrator.EXAMPLE>nfkminfo
World
generation 2
state 0x3737000c Initialised Usable ...
...
Module #1
generation 2
state 0x2 Usable
...
```

3.3. Create the OCS or Softcard

If using OCS protection, create the OCS now. Follow your organization's security policy for the value N of K/N. As required, create extra OCS cards, one for each person with access privilege, plus spares.



Administrator Card Set (ACS) authorization is required to create an OCS in FIPS 140 level 3.



After an OCS card set has been created, the cards cannot be duplicated.

- 1. If using remote administration, ensure the C:\ProgramData\nCipher\Key Management Data\config\cardlist file contains the serial number of the card(s) to be presented.
- 2. Open a command window as Administrator.
- Run the following command. Follow your organization's security policy for the values K/N. The OCS cards cannot be duplicated after created. Enter a passphrase or password at the prompt. Notice that slot 2, remote via a Trusted Verification Device (TVD), is used to present the card. In this example, K=1 and N=1.

```
>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: empty
   Module 1 slot 3: empty
   Module 1 slot 2: blank card
   Module 1 slot 2:- passphrase specified - writing card
Card writing complete.
cardset created; hkltu = a165a26f929841fe9ff2acdf4bb6141c1f1a2eed
```

Add the -p (persistent) option to the command above to retain authentication after the OCS card has been removed from the HSM front panel slot, or from the TVD. If using OCS card protection and the non-persistent card configuration, OCS cards need to be inserted in the nShield front panel or always present in the TVD. The authentication provided by the OCS as shown in the command line above is nonpersistent and only available for K=1 and while the OCS card is present in the HSM front panel slot or TVD.

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 a165a26f929841fe9ff2acdf4bb6141c1f1a2eed 1/1 none-NL testOCS

The **rocs** utility also shows the OCS 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 1
rocs> quit

If using Softcard protection, create the Softcard now.

1. 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. Run the following command and enter a passphrase/password at the prompt:

```
>ppmk -n testSC
Enter new pass phrase:
Enter new pass phrase again:
New softcard created: HKLTU d9414ed688c6405aab675471d3722f8c70f5d864
```

3. Verify the Softcard was created:

```
>nfkminfo -s
SoftCard summary - 1 softcards:
Operator logical token hash name
d9414ed688c6405aab675471d3722f8c70f5d864 testSC
```

The **rocs** utility also 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 1
2 testSC 0 (0) (softcard)
rocs>quit
```

3.4. Install and register the CNG provider

To install and register the CNG provider:

- 1. Select Start > Entrust > CNG configuration wizard.
- 2. Select Next on the Welcome window.

nShield CNG Providers Config	guration Wizard	
ENTRUST	Welcome to the nShield support software This wirad guides you through the installation of nShield's Cyplographic Service Providers. NShield CNG Providers enable the use of nShield modules with the wirde range of security-enabled applications provided with Windows. If you have not already created an nShield security world or a suitable card set, the wizard guides you through their creation before registreing the CNG Providers. To continue, click Next.	
	< Back Next > Cance	el

 Select Next on the Enable HSM Pool Mode window, leaving Enable HSM Mode for CNG Providers un-checked.



If you intend to use multiple HSMs in a failover and load-sharing capacity, select **Enable HSM Pool Mode for CNG Providers**. If you do, you can only use module protected keys. Module protection does not provide conventional 1 or 2 factor authentication. Instead, the keys are encrypted and stored as an application key token, also referred to as a Binary Large Object (blob), in the kmdata/local directory.

- 4. Select Use existing security world on the Initial setup window. Then select Next.
- 5. Select the HSM (Module) if more than one is available on the **Set Module States** window. Then select **Next**.

Ensure module	t es es are in the correct s	tate before you proceed.	ENTR
The following	modules are available	e in your system:	
Module ID	Mode	State	
1	operational	usable	
2	operational	usable	
At least one m Or reset modu uninitialized nº Refer to the u state. If you n restart the wiz	odule is usable in the le 2 to the initializatio Shield modules. ser guide for details o eed to power down y ard on boot up to cor	current world. Click Next t n state to enable you to res f how to put your nShield m our computer, select the tic titinue the installation.	o continue with this world. tore your security world to nodule in the initialization kbox below and then

6. In Key Protection Setup, select Operator Card Set protection. Then select Next.



7. Choose from the **Current Operator Card Sets** or **Current Softcards** list. These were created above. Then select **Next** and **Finish**.

Х
, or create a new token.
t Token Information: testOCS 0xa165a26f ers: 1 of 1, Non-persistent None ing: none
I otal number of cards (N):

8. Verify the provider with the following commands:

>certutil -csplist findstr nCipher
Provider Name: nCipher DSS Signature Cryptographic Provider
Provider Name: nCipher Enhanced Cryptographic Provider
Provider Name: nCipher Enhanced DSS and Diffie-Hellman Cryptographic Provider
Provider Name: nCipher Enhanced DSS and Diffie-Hellman SChannel Cryptographic Provider
Provider Name: nCipher Enhanced RSA and AES Cryptographic Provider
Provider Name: nCipher Enhanced SChannel Cryptographic Provider
Provider Name: nCipher Signature Cryptographic Provider
Provider Name: nCipher Security World Key Storage Provider
>cnglist.exe list-providers findstr nCipher nCipher Primitive Provider nCipher Security World Key Storage Provider

9. Check the registry in CNGRegistry:

HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Control\Cryptography\Providers\nCipherSecurityWorldKeyStorageProvid er

📑 Registry Editor				_		×
File Edit View Favorites Help						
Computer\HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Control\Crypt	ograp	hy\Providers\nCipl	ner Security World Ke	y Storage F	Provider	
✓	^	Name	Туре	Da	ta	
Microsoft Key Protection Provider		(Default)	REG SZ	(va	lue not se	t)
> 📜 Microsoft Passport Key Storage Provider			-			
>] Microsoft Platform Crypto Provider						
>] Microsoft Primitive Provider						
Microsoft Smart Card Key Storage Provider						
> I Microsoft Software Key Storage Provider						
Microsoft SSL Protocol Provider						
In Cipher Primitive Provider						
Incipher Security World Key Storage Provider						
Windows Client Key Protection Provider						
> 📙 WebSignIn						

3.5. Install and configure SqlServer PowerShell module

1. Open a PowerShell session as Administrator and run:

```
PS C:\Users\Administrator.EXAMPLE> [Net.ServicePointManager]::SecurityProtocol =
[Net.SecurityProtocolType]::Tls12
PS C:\Users\Administrator.EXAMPLE> Install-PackageProvider Nuget -force -verbose
VERBOSE: Acquiring providers for assembly: C:\Program
Files\WindowsPowerShell\Modules\PackageManagement\1.4.7\fullclr\Microsoft.PackageManagement.CoreProviders.d
ll
...
VERBOSE: Imported provider 'C:\Program
Files\PackageManagement\ProviderAssemblies\nuget\2.8.5.208\Microsoft.PackageManagement.NuGetProvider.dll' .
```

2. Update PowerShellGet:

PS C:\Users\Administrator.EXAMPLE> Install-Module -Name PowerShellGet -force -verbose
VERBOSE: Using the provider 'PowerShellGet' for searching packages.
...
VERBOSE: Module 'PowerShellGet' was installed successfully to path 'C:\Program
Files\WindowsPowerShell\Modules\PowerShellGet\2.2.5'.

3. Download and install the SqlServer module to configure Always Encrypted using Power Shell:

```
PS C:\Users\Administrator.EXAMPLE> Install-Module -Name SqlServer -force -verbose -AllowClobber
VERBOSE: Using the provider 'PowerShellGet' for searching packages.
...
VERBOSE: Module 'SqlServer' was installed successfully to path 'C:\Program
Files\WindowsPowerShell\Modules\SqlServer\21.1.18256'.
```



The -AllowClobber parameter allows you to import the specified commands if it exists in the current session.

4. Once installed, confirm the install by running the command below.



If you are using PowerShell ISE, refresh the Commands pane. If you are using PowerShell, open a new session.



3.6. Install the SQL Server Management Studio

Install the SQL Server Management Studio.

3.7. Allow Active Directory user to remote login

To allow an Active Directory user to remote login:

- 1. Select Control Panel > System > Advance system settings.
- 2. Select the Remote tab in the System Properties dialog. Then select Select Users....
- 3. Add the following users:

ī.

- ° <domain>\dbuser
- ° <domain>\dbuser2.

System Propertie	S					
Computer Name	Hardware Adv	anced	Remote			
Remote Deskt	op Users				?	\times
The users listed the Administrat	l below can conr ors group can co	nect to t	this comp even if the	uter, and a ey are not l	any membe isted.	ers of
	lbuser lbuser2			-		
Adm	inistrator already	/ has ac	ccess.			
Add	Remove					
To create new u Panel and oper	user accounts or User Accounts	r add us	sers to oth	ner groups	, go to Cor	ntrol
				ОК	Can	cel
		ОК		Cancel		Apply

Chapter 4. Install and configure SQL server

This installation must be performed on the remote server.

Installation steps:

- 1. Install the SQL database engine.
- 2. Create the SQL logins.

4.1. Install the SQL database engine

This installation must be performed on the remote server using the <domain_name>\Administrator account.

- 1. Install the SQL engine.
- 2. Open the firewall ports 1433, 1434, and 445 for access by the SQL database engine, SQL browser, and Active Directory for domain account authorization.

💣 Windows Defender Firewall with	h Advanced Security											-
File Action View Help												
🗢 🏟 🙍 🖬 🗟 🚺												
Pindows Defender Firewall with	Inbound Rules											
Inbound Rules	Name	Group	Profile	Enabled	Action	Override	Program	Local Address	Remote Address	Protocol	Local Port	Remote Port ^
Connection Security Rules	🤨 Firefox (C:\Program Files\Mozilla Firefox)		Private	Yes	Allow	No	C:\Progr	Any	Any	UDP	Any	Any
> 🖳 Monitoring	🔮 Firefox (C:\Program Files\Mozilla Firefox)		Private	Yes	Allow	No	C:\Progr	Any	Any	TCP	Any	Any
	MS-SQL-Browser-Server		All	Yes	Allow	No	Any	Any	Any	TCP	1434	Any
	🔮 MS-SQL-Database-Engine		All	Yes	Allow	No	Any	Any	Any	TCP	1433	Any
	🖉 MS-SQL-SSMS		All	Yes	Allow	No	Any	Any	Any	TCP	445	Any
	🔇 Rule to fix RPC Server Unavailable Error		All	Yes	Allow	No	c:\Wind	Any	Any	TCP	RPC Endp	Any
	🔇 sqlserver		All	Yes	Allow	No	Any	Any	Any	TCP	1433	Any
	🧭 AllJoyn Router (TCP-In)	AllJo_	Dom	Yes	Allow	No	%Syste	Any	Any	TCP	9955	Any
	🧭 AllJoyn Router (UDP-In)	AllJo_	Dom	Yes	Allow	No	%Syste_	Any	Any	UDP	Any	Any

3. Create a test database, if a suitable is not available, for the purpose of this integration.

SQLQuery1.sql - MS-SQL-AE-Srvcom.TestDatabase (\Admir	nistrator (57)) - Microsof	t SQL Server Management	Quick Launch (Ctrl+Q)	_ م		×
File Edit View Query Project Tools Window Help								
💿 - 💿 🎨 - 'n - 😩 💾 🚰 🖨 New Query 🗯 🖓 🔬		¥ 🗇 🖞	1 9 - 1			- 🗔 🌶 🕯	i 🖂 🗸	÷
🕴 🕆 🕅 TestDatabase 🔹 🕨 Execute 🔳 🗸 🛱 🗐		80 80 🗊		🖸 🗉 🖆 🝜 🚈 🗞	÷			
Object Explorer 🗸 🗘 🗙	SQLO	Query1.sql -	MSAdmin	istrator (57)) 😕 🗙				•
Connect - 🛱 🎽 🔳 🝸 🖒 🔸		/*****	Script f	or SelectTopNRows comm	nand from SSMS *	*****/		÷
R MS SOL AE Source (SOL Service 15.0		SELECT	TOP (1000) [FirstName]				1
Databases		ر	LastName	1				
Databases		,	[Email] [Password	1				
System Databases		FROM	TestData	J basel.[dbol.[TestTable	-1			
					-			1
Database Diagrams								
T System Tables								
EleTables								
External Tables								
Graph Tables	100 %	6 - 4					P	
III III dho TestTable	R	esults _® ¥ Me	ssages					
		FirstName	LastName	Email	Password			^
Evenal Resources	1	Jack	Snepard	jack.shepard@testserver.com	%#[Bq1,z4B&_UM5			
	2	Kate	Austin	john.locke@itestserver.com	Il&wbaca85 #II			
B Programmability	4	James	Ford	iames ford@testserver.com	J5YPbd59w\$5siuk			
	5	Ben	Linus	ben.linus@testserver.com	MY1=q=&qm{.UATC			
	6	Desmond	Hume	desmon.hume@testserver.com	aPoTEp)h;TfNWT1			
	7	Daniel	Faraday	daniel.faraday@testserver.com	9MPDzVhXY]S]Q%%			
E Security	8	Sayid	Jarrah	sayid.jarrah@testserver.com	GfonIxi][H{m9w}			
Security	9	Richard	Alpert	richard.alpert@testserver.com	!btA9LSRUgsttRH			\checkmark
V Server objects			Smith	racon smith@testsenver.com	Hereinistrator TestDa	tabasa 00:00:00	10	
		ie 🖬 MS	-SQL-AE-Sr	vcom (\ \A	aministrator lestDa	itabase 00:00:00	TO FOW	s
☐ Ready Ln 6 Col 31	Ch			NS				

4.2. Create the SQL logins

1. Create two SQL logins with the domain accounts <domain>\dbuser and <domain>\dbuser2 with **Default Database** equal to "TestDatabase".



2. Set the User Mapping as database owners, db_owner, of TestDatabase.

Login Properties -	\dbuser		- 1	⊐ ×				
Select a page & General								
 Server Roles User Mapping 								
 Securables Status 	Map Database master model msdb	User	Default Schema					
Г	✓ TestDatabase	dbuser	dbo					
Connection								
Server:	Guest account enabled for:	master						
MS-SQL-AE-SRV Connection:	Database role membership for:	master						
Administrator	db_accessadmin db_backupoperator db_datareader db_datawriter db_ddladmin db_ddladmin db_denydatareader							
Progress	db_denydatawriter							
C Ready	db_securityadmin							
			ОК	Cancel				

Chapter 5. Generate the encryption keys

To generate encryption keys:

- Generate the Always Encrypted Column Master Key (CMK).
- Generate My Column Master Key (MyCMK) and My Column Encryption Key (MyCEK) with SSMS.
- Generate MyCMK and MyCEK with PowerShell.

5.1. Generate the Always Encrypted Column Master Key (CMK)

The CMK is protected by the nShield HMS.

- 1. Log in to the client using the <domain>\dbuser, or a suitable security administrator account.
- 2. Launch PowerShell and run the Generate_AECMK.ps1 script (shown below).

```
$cngProviderName = "nCipher Security World Key Storage Provider"
$cngAlgorithmName = "RSA"
$cngKeySize = 2048
$cngKeyName = "AECMK"
$cngKeyName = "AECMK"
$cngKeyParams = New-Object System.Security.Cryptography.CngProvider($cngProviderName)
$cngKeyParams.revoider = $cngProvider
$cngKeyParams.revoider = $cngProvider
$cngKeyParams.KeyCreationOptions =
[System.Security.Cryptography.CngProperty("Length",
[System.BitConverter]::GetBytes($cngKeySize), [System.Security.Cryptography.CngPropertyOptions]::None);
$cngKeyParams.Parameters.Add($keySizeProperty)
$cngAlgorithm = New-Object System.Security.Cryptography.CngAlgorithm($cngAlgorithmName)
$cngKey = [System.Security.Cryptography.CngAlgorithm, $cngKeyName, $cngKeyParams)
$cngKeyParams.Parameters.Parameters.Parameters.Parameters.Parameters.Parameters.Parameters.Parameters.Parameters.Parameters.Parameters.Parameters.Parameters.Parameters.Parameters.Parameters.Parameters.Parameters.Parameters.Parameters.Parameters.Parameters.Parameters.Parameters.Parameters.Parameters.Parameters.Parameters.Parameters.Parameters.Parameters.Parameters.Parameters.Parameters.Parameters.Parameters.Parameters.Parameters.Parameters.Parameters.Parameters.Parameters.Parameters.Parameters.Parameters.Parameters.Parameters.Parameters.Parameters.Parameters.Parameters.Parameters.Parameters.Parameters.Parameters.Parameters.P
```

a. Run the following command:

> PowerShell -ExecutionPolicy Bypass -File Generate_AECMK.ps1

The following dialog appears.

÷	nCipher Key Storage Provider - Create key	×
	Create new key:	
	AECMK	
	<u>N</u> ext Cance	I

- b. Select Next.
- c. Select the **Operator Card Set Protection**. Insert the OCS card in the HSM and select **Next**.



d. Select the OCS and then Select **Next**.

				×
÷	nCipher Key Storage Provider	r - Create key		
	Select token to protect th	e key with.		
	Current Operator Card Sets:	Operator Card Set Tok Name: Token hash: Sharing parameters: Timeout: Currently protecting:	en Information: testOCS 0xa165a26f 1 of 1, Non-persistent None none	
			<u>N</u> ext Canc	el

e. Select the HSM and select **Finish**.

				\times
←	nCipher Key Storage Provider	- Create key		
	Choose modules you wis	h to load th	ne key onto.	
	Excluded modules:		Included modules:	
	Module #2	Add Remove	Module #1	
		Add all		
		Remove all		
	You may not use more than 1 module non-persistent cards and comprises	e, because the only of 1 card.	card set you have chosen has	
			<u>Finish</u> Can	cel

f. Enter the OCS passphrase and select **Next**.

		\times
~	nCipher Key Storage Provider	
	Module 1 slot 2: 'testOCS' #1	
	You must enter a passphrase for this card	

	Next	ancel
	INEXL	ancel

g. Select Finish.

Card rea	ding com	plete.		
Module	Slot	Content	Status	
1	5		complete	
1	4		complete	
1	2		complete	
1	0		complete	

A 2048-bit RSA key pair, called AECMK, has been generated. The key is encrypted in the HSM and then pushed to the requesting client server, where it is stored as an Application Key Token in the %NFAST_KMDATA%\local folder. That is, :\ProgramData\nCipher\Key Management Data\local.

3. Verify the new key:

C:\Users\Administrator.EXAMPLE>nfkminfo -k Key list - 1 keys AppName caping Ident user--e57798f862740453d02379579c1758ddfa2189db

4. Display the information about the key by copy-pasting the key name above as follows:

C:\Users\Administrator Key AppName caping Ide BlobKA length BlobPubKA length BlobRecoveryKA length name hash recovery protection other flags cardset gentime SEE integrity key	EXAMPLE>nfkminfo -k caping usere57798f862740453d02379579c1758ddfa2189db nt usere57798f862740453d02379579c1758ddfa2189db 1128 484 1496 "AECMK" d9253d650283dafd8d62659f9fb74102b9edcf8c Enabled CardSet PublicKey !SEEAppKey !NVMemBlob +0x0 a165a26f929841fe9ff2acdf4bb6141c1f1a2eed 2022-12-30 19:46:54 NONE
BlobKA	
format	6 Token
other flags	0x0
hkm	28ee9f7cfceba95992f1f3f31b39c8dba7cfa960
hkt	a165a26f929841fe9ff2acdf4bb6141c1f1a2eed
hkr	none
BlobRecoveryKA	
, format	9 UserKey
other flags	0x0
hkm	none
hkt	none
hkr	55c38c84103d95278fd54b6b5b3e67d614db8538
BlobPubKA	
format	5 Module
other flags	0x0
hkm	c2be99fe1c77f1b75d48e2fd2df8dffc0c969bcb
hkt	none
hkr	none
Extra entry #1 typecode length Not a blob	0x10000 65536 60

5.2. Generate My Column Master Key (MyCMK) and My Column Encryption Key (MyCEK) with SSMS

This key will encrypt all subsequent Column Encryption keys (CEKs) in your database.

- 1. Log in to the client using the <domain>\dbuser account.
- 2. Launch Microsoft SQL Server Management Studio.
- 3. Connect to the database on the remote SQL server:
 - a. Select the **Login** tab and set it as follows:

🖵 Connect to Server		\times							
	SQL Server								
Login Connection Properties Always Encrypted Additional Connection Parameters									
Server									
Type the server name, or cho	oose it from the drop-down list.								
Server type:	Database Engine	\sim							
Server name:	MS-SQL-AE-Srvcom	~							
Authentication:	Windows Authentication	\sim							
User name:	\dbuser	~							
Password:									
	Remember password								
Co	Cancel Help Op	uons <<							

b. Select the **Connection Properties** tab, as set as follows:

Connect to Server		\times
	SQL Server	
Login Connection Propertie	Always Encrypted Additional Connection Parameters	
Type or select the name of t	the database for the connection.	
Connect to database:	<default></default>	\sim
Network		
Network protocol:	<default></default>	\sim
Network packet size:	4096 🗢 bytes	
Connection		
Connection time-out:	30 seconds	
Execution time-out:	0 seconds	
Encrypt connection		
✓ Trust server certificat	te	
Use custom color:	Select	
	Reset All	
	Connect Cancel Help Options	<<

c. Select the Always Encrypted tab and select Enable Always Encrypted:

모 ^를 Connect to Server	×
SQL Server	
Login Connection Properties Always Encrypted Additional Connection Parameters	
Second text Interval (Column encryption)	
Enclave Attestation URL:	
Type the URL for attesting the server-side enclave, if y are using Always Encrypted with secure enclaves.	ou
Learn	More
Connect Cancel Help Option	s <<

- d. Select Connect.
- Using the Object Explorer, select the Security directory under the required database, then select Always Encrypted Keys > Column Master Key > New Column Master Key.

🔀 SQLQuery1.sql - MS-SQ	2L-AE-Srv.interop.com.TestDatabase (\dbuse	r (55)) - I	Microsoft SO	QL Server M	anagement Studio	Quick Lau	nch (Ctrl+Q)	٩	-		х
File Edit View Project	Tools Window Help										
80-0 -		VE		_ @ _			_		6		
					°□ * <mark>^</mark>		•		~ -		* -
§₩ ¥	🔹 🕨 Execute 🔳 🗸 🗄 🗐				°≣ <u>≈ ≥</u> *3	₽ =					
Object Explorer	▲ Å ×	SQLC)uery1.sql -	M \(dbuser (55)) 🛛 😕 🗙						•
Connect 👻 🍟 🎽 🝸	C -*		/****** SELECT 1	Script f	or SelectTopNR	ows comm	nand from SSMS	*****	*/		+
🖃 🗟 MS-SQL-AE-Srv	om (SQL Server 15.0. \dbuser) 🔨		,	LastName	·]						
🖃 📁 Databases			,	[Email]	-						
🖽 ≡ System Databases				[Password	1]						
🗄 ≡ Database Snapshots	5		FROM	[TestData	base].[dbo].[T	estTable	2]				
🖃 🗎 TestDatabase											
🗄 🖷 Database Diagram	ns										
🖽 💻 Tables											
🗄 💻 Views											
🗄 💻 External Resources	s										
🗄 💻 Synonyms											_
🗄 💻 Programmability		100.9								h	
🖽 💻 Service Broker		100 %	oulto mas							,	
🗉 💻 Storage			esuits B# Me	ssages	F N		D				_
🖃 💻 Security			FirstName	LastName	Email jock shopprd@tostsr	onior com	%#IBoT 74B8 LIM5				
🕀 📁 Users		2	John	Locke	jack.sneparu@testsen/	er com	v@2Myr:XYcYsIPw				
🗄 📁 Roles		3	Kate	Austin	kate.austin@testserv	ver.com	I!8wbaca85 #If				
🗄 📁 Schemas		4	James	Ford	james.ford@testserv	er.com	J5YPbd59w\$5siuk				
🗄 📁 Asymmetric Key	∕S	5	Ben	Linus	ben.linus@testserver	r.com	MY1=g=&gm{.UATC				
🗄 🖷 Certificates		6	Desmond	Hume	desmon.hume@test	server.com	aPoTEp)h;TfNWT1				
🗄 🖷 Symmetric Keys	5	7	Daniel	Faraday	daniel.faraday@tests	server.com	9MPDzVhXYJSJQ%%				
= Always Encrypte	ed Kevs	8	Sayid	Jarrah	sayid.jarrah@testser	ver.com	Gfonlxi][H{m9w}				
🗄 ≡ Column M	New Column Master Key	9	Richard	Alpert	richard.alpert@testse	erver.com	IbtA9LSRUgsttRH				
🗄 💻 Column En		10	Jacob	Smith	jacob.smitn@testser	ver.com	EZg4[Id)NVVE=D;				
🗄 💻 Database Au	Start PowerShell	Q	MS-SQ	L-AE-Srv.	.com (\dbuse	er (55) TestDatabase	00:00	0:15	10 row	s
	Keports										
L/ Ready	Refresh										

- 5. Enter the following information on the **Column Master Keys** dialog:
 - a. Enter a **Name**, for example **MyCMK**.
 - b. Select **Key Storage Provider (CNG)** from the **Key store** drop-down list and then **Select a provider**.
 - c. Select nCipher Security World Key Storage Provider from the drop-down list.

The AECMK key created in an earlier step appears in Name.

d. Select **OK** to create a new key using the nShield HSM and CNG KSP.

🗝 New Column Master Key				—		\times
Select a page	🗊 Script 🔻 😨) Help				
	Name:	МуСМК				
	Key store:	Key Storage Provider (CNG)	~	Refresh		
	Select a pro	vider:				
	Name					
	AECMK					
Connection						
Server: MS-SQL-AE-SRV Connection: dbuser						
View connection properties						
Progress						
C Ready						
	Gen	erate Key				
			[ОК	Ca	ncel

6. Select Next.

The newly-created **MyCMK** is created in the database under **Security** > **Always Encrypted Keys** > **Column Master Keys**.



7. Using Object Explorer, select the Security directory under the required database.

Select Always Encrypted Keys to expand it, then select New Column Encryption Key.

8. Enter **Name**, select the CMK, then select **OK**.

🔐 New Column Encryption Ke	у		_		\times
Select a page	🗊 Script 🔻 😧 Help				
	Name: Column master key: Column encryption k encryption keys. This To create a new colu	MyCEK MyCMK keys protect your data, and column master keys lets you manage fewer keys. umn master key, use the "New Column Master H	protect your colu Key" page.	Refresh	
Connection					
Server: MS-SQL-AE-SRV Connection: \dbuser \Usew connection properties					
Progress					
C Ready					
			ОК	Can	cel

9. Present the OCS and then select **Next**.

	×
 nCipher Key Storage Provider - Load key 	
Load key:	
	Next Cancel

10. Select the HSM and then select **Finish**.

 \times

← nCipher Key Storage Provider - Load key

Choose modules you wish to load the key onto.

Add Remove	Module #1		
Remove			
Add all			
Remove all			
	Add all Remove all e, because the ca	Add all Remove all b, because the card set you ha	Add all Remove all b, because the card set you have chosen has not of 1 card

11. Enter the passphrase and then select **Next**.

		\times
←	nCipher Key Storage Provider	
	Module 1 slot 2: 'testOCS' #1 You must enter a passphrase for this card	
	•••••	
	Next Cance	el

12. Select **Finish** after the OCS card reading completes.

Module				
1	Slot	Content	Status	
1	4		complete	
1	3		complete	
1	2		complete	
1	0		complete	

The newly-created **MyCEK** is in the database under **Security** > **Always Encrypted Keys** > **Column Encryption Keys**.



5.3. Generate MyCMK and MyCEK with PowerShell

To generate MyCMK and MyCEK with PowerShell:

- 1. Delete MyCEK and MyCMK in that order created above by right-clicking each key and selecting **Delete**.
- 2. Launch PowerShell and run the Generate_MyCMK_and_MyCEK.ps1 script (below).

```
# Import the SqlServer module.
Import-Module SqlServer
# Connect to database.
$ConnectionString = "Data Source=MS-SQL-AE-Srv.interop.com,1433;Initial
Catalog=TestDatabase;Trusted_Connection=True;MultipleActiveResultSets=False;Encrypt=True;TrustServerCertifi
cate=True;Packet Size=4096;Application Name=` "Microsoft SQL Server Management Studio`""
$Database = Get-SqlDatabase -ConnectionString $ConnectionString
# Create a SqlColumnMasterKeySettings object for your column master key.
$cmkSettings = New-SqlCngColumnMasterKeySettings -CngProviderName "nCipher Security World Key Storage
Provider" -KeyName "AECMK"
# Create column master key metadata in the database.
New-SqlColumnMasterKey -Name "MyCMK" -InputObject $Database -ColumnMasterKeySettings $cmkSettings
# Generate a column encryption key, encrypt it with the column master key and create column encryption key
metadata in the database.
New-SqlColumnEncryptionKey -Name "MyCEK" -InputObject $Database -ColumnMasterKey "MyCMK"
```

The command line is:

> PowerShell -ExecutionPolicy Bypass -File Generate_MyCMK_and_MyCEK.ps1

Name

МуСМК	 	
МуСЕК		

- 3. Present the OCS, select the HSM, and enter the passphrase.
- 4. Check the newly-created **MyCMK** and **MyCEK** are present.

Chapter 6. Encrypt or decrypt a column with SSMS

To encrypt or decrypt a column with SSMS:

- Encrypt a column
- View an encrypted column
- Remove column encryption

6.1. Encrypt a column

- 1. Log in to the client with the <domain>\dbuser account.
- 2. Launch Microsoft SQL Server Management Studio.
- 3. Connect to the database on the remote SQL server, enabling **Always Encrypted**, see [encrypt-decrypt-column-with-ssms:::generate-mycmk-mycek-ssms].
- In the Object Explorer, right-click the TestDatabase database and select Tasks > Encrypt Columns....
- 5. On the Introduction screen, select Next.

個 Always Encrypted	- 0	×
Introduction	0) Help
Column Selection		
Master Key Configuration		
Run Settings	Always Encrypted is designed to protect sensitive information - such as credit card numbers - store SQL Server databases. It enables clients to encrypt data inside client applications and never reveal th	d in Ie
Summary	encryption keys to SQL Server.	
Results		
	•	
	ii	
	·/	
	Do not show this page again.	
	< Previous Next > Can	cel

6. On the **Column Selection** screen, select the column **Name**, **Encryption Type**, and **Encryption Key**. Then select **Next**.

					- 🗆	>
Column Selection						
troduction					0	Help
olumn Selection						
laster Key Configuration	Search column name					
un Settings	Apply one key to all che	cked columns:		MyC	EK	\sim
ummary			Encryption Type	()	Encryption Key	()
esults	Name	State	Encryption Type		Encryption Key	
	dbo.TestTable					
	Password	I	Randomized	•	MyCEK	٠
	Show affected columns	only				

7. On the Master Key Configuration screen, select Next.

团 Always Encrypted		-		×
Master Key Configu	ration			
Introduction Column Selection Master Key Configuration Run Settings Summary Results	No additional configuration is necessary because you are using existing keys.		Ø	Help
	< Previous Ne	t>	Cance	ł

8. On the Run Settings screen, select Proceed to finish now. Then select Next.

翻 Always Encrypted	– 🗆 X
Introduction Column Selection Master Key Configuration Run Settings Summary Results	While encryption/decryption is in progress, write operations should not be performed on a table. If write operations are performed, there is a potential for data loss. It is recommended to schedule this encryption/decryption operation during your planned maintenance window.
I	Select how you would like to proceed O Generate PowerShell script to run later O Proceed to finish now

9. On the **Summary** screen, verify the configuration choices. Then select **Finish**.

钮 Always Encrypted	_	-		×
Summary				
Introduction Column Selection Master Key Configuration Run Settings	Verify the choices made in this wizard. Click Finish to perform the operations with the following settings:		@	Help
Summary Results	Source database settings Source database name: MS-SQL-AE-Srv. com Source database name: TestDatabase Encrypt column Password Table name: TestTable Encryption key name: MyCEK Encryption type: Randomized		Capital	
	< Previous Finish		Canc	el

- 10. Present the OCS, select the HSM, and enter the passphrase.
- 11. Check that **Passed** appears in the **Details** column of the **Results** screen.

翻 Always Encrypted			-	o ×
Results				
Introduction				Help
Column Selection				
Master Key Configuration				
Run Settings				
Summary	Summary:			_
Results	Task		Details	
	Performing encryption operations		Passed	
	Always Encrypted Wizard Log Report			
		< Previous Nex	d >	Close



The column is encrypted in the SQL server, but it shows as clear text on the **Microsoft SQL Server Management Studio** GUI on the client. This is because **Always Encrypted** is performing the decryption at the client site.

12. Select Close.

6.2. View an encrypted column

Reconnect to the SQL server with **Enable Always Encrypted** disabled to view the encrypted data stored in the SQL server.

1. Connect to the SQL server but with the **Enable Always Encrypted** unchecked.

🖵 Connect to Server	\times
SQL Server	
Login Connection Properties Always Encrypted Additional Connection Parameters	
Enable Always Encrypted (column encryption)	
Enclave Attestation URL:	
Type the URL for attesting the server-side enclave, if you are using Always Encrypted with secure enclaves.	
Learn Mo	re
Connect Cancel Help Ontions <	<
Control Control	

2. Right-click **dbo.Table** and select **Select Top 1000 Rows**. The column that was chosen for encryption now appears as ciphertext, that is, as an encrypted value.



- 3. Reconnect to the SQL server, but with the Enable Always Encrypted checked.
- 4. Present the OCS, select the HSM, and enter the passphrase.
- Right-click dbo.Table and select Select Top 1000 Rows. The column that was chosen for encryption is now being decrypted by Always Encrypted with the key protected by the nShield HSM.

🔀 SQLQuery3.sql - MS-SQL-AE-Srvcom.TestDatabase (\dbuser	(64)) - Mic	crosoft SQL	Server M	anagement Studio Quick Lau	inch (Ctrl+Q)	₽ -		×
File Edit View Query Project Tools Window Help								
🗴 😋 🔹 💿 🐘 🔹 🐂 🗳 👘 New Ouery 🕼 💭 💭 💭	жПб	ຄ∣୭ -	C - 1	3 - 5	-		= ► [] -	
								Ŧ
InterstDatabase	- 10 B-			⊴ =≤ ≥= २० –				
Object Explorer 🔹 👎 🗙	SQLQue	ery3.sql - M.	\o	dbuser (64)) ≄ × SQLQuer	y2.sql - M \dbu	ıser (68))		Ŧ
Connect 👻 🍟 🌹 🍸 🖒 🦘	/* 	****** S	cript f p (1000	or SelectTopNRows comr) [FirstName]	nand from SSMS *	*****/		÷
🗆 📾 MS-SQL-AE-Srvcom (SQL Server 15.0. \dbuser) \land		,[Li	astName]			1	-
🖃 🗯 Databases		, [Er	mail]	-				
🗄 🖷 System Databases		, [Pa	assword]				
🗄 🖷 Database Snapshots		FROM [T	estData	base].[dbo].[TestTable	e]			
🗆 🗑 TestDatabase								
🗄 🖷 Database Diagrams								
🖂 🛲 Tables								
🗄 💻 System Tables								
🗉 💻 FileTables								
🗄 📁 External Tables								
🗄 📁 Graph Tables	100.94							
⊞ III dbo.TestTable	IOU 70	Ho - T Marca						
🗄 💻 Views		its gr Messa	ages	E a al	Description	1		
🗄 💻 External Resources		irstname L	astname	Email	%#fBaT z4B& LIM5			
🗄 💻 Synonyms	2 .4	ohn I	ocke	john locke@testserver.com	v@2Myr:XYcYsIPw			
🗄 🖷 Programmability	3 K	ate A	ustin	kate.austin@testserver.com	1!8wbaca85 #1[
🗄 💻 Service Broker	4 Ja	ames F	ord	james.ford@testserver.com	J5YPbd59w\$5siuk			
🗄 🖷 Storage	5 B	Ben L	inus	ben.linus@testserver.com	MY1=g=&gm{.UATC			
H = Security	6 D	Desmond H	lume	desmon.hume@testserver.com	aPoTEp)h;TfNWT1			
III Security	7 D	Daniel F	araday	daniel.faraday@testserver.com	9MPDzVhXY]S]Q%%			
I Server Objects	8 S	Sayid J	arrah	sayid.jarrah@testserver.com	Gfonlxi][H{m9w}			
Benlication	9 R	Richard A	lpert	richard.alpert@testserver.com	!btA9LSRUgsttRH			
B PolyBase	10 Ja	acob S	mith	jacob.smith@testserver.com	EZg4[Id)NWvE=D;			
🗄 = Always On High Availability		MC COL /		and Alburg	(6.4) TestDatabase	00-00-10	10	-1
- · · · · · · · · · · · · · · · · · · ·		IVIS-SQL-A	AE-SIV	.com (\dbuse	er (64) lestDatabase	00:00:10	TO FOWS	
🗖 Ready Ln 1 Col 1 Ch 1			NS					

6.3. Remove column encryption

1. In the **Object Explorer**, right-click the **TestDatabase** database, and select **Tasks** > **Encrypt Columns...**.

🔀 SQLQuery3.sql -	MS-SQL-AE-Srv.interop.com.TestDatabase (\dbuser (64)) - Microsoft SQL Server Management St	udio Quick Lau	nch (Ctrl+Q)	٩	-		×
File Edit View	File Edit View Project Tools Window Help							
0 • 0 📸 • 1	🛅 👻 🛀 🔐 💭 New Query 🖉	Detach		-		۵ 🖻		•
🕴 👻 🕴 TestDatab	ase - Execute	Take Offline	*@ _					
Object Surfaces		Bring Online	1 X 501000	V2 col M i) d	burgar 169	m		-
Object Explorer	- • •	Stretch •	ppNRows com	yz.sql-m \d and from SSMS	******	/ /		÷
Connect 🕶 🌹 🐂 🗏	T C **	Encrypt Columns	ne]					-
B MS-SQL-AE-Srv.	.com (SQL Server 15.0.	Data Discovery and Classification						
Databases	22505	Vulnerability Assessment						ш
🗄 💻 Database Sna	pshots	Shrink].[TestTable	•]				ш
🖃 🗑 TestDatab		Back Up						ш
🗄 🖷 Databas	New Database	Restore						ш
🖯 🛲 Tables	New Query	Mirror	-					-
🗄 📫 Syste	Script Database as	Launch Database Mirroring Monitor						
🗄 🖷 File I a	Tasks •	Ship Transaction Logs						
Extern	Policies •	Generate Scripts						Ŧ
⊞ ≡ dho 1	Facets	Generate In-Memory OLTP Migration Checklists					•	
🗄 📁 Views	Start PowerShell	Extract Data-tier Application						
🗄 🖷 External	Azure Data Studio	Deploy Database to Microsoft Azure SOL Database	testsener com	%#fBgT z4B& LIM5				
🗄 💻 Synonyi	Azure SQL Managed Instance link	Export Data-tier Application	stserver.com	v@2Mbr;XYcYsIPw				
🕀 📁 Program	Reports +	Register as Data-tier Application	stserver.com	I!8wbgcg85#I[
🗄 📫 Service	Rename	Upgrade Data-tier Application	stserver.com	J5YPbd59w\$5siuk				
🗄 📫 Storage	Delete	Delete Data-tier Application	atestserver.com	aPoTEp)h:TfNWT1				
E Security	Refrech	Import Flat File	ptestserver.com	9MPDzVhXY[S]Q%%				
E Server Object	Properties	Import Data	estserver.com	Gfontxi][H{m9w}				
Benver Objec Benver Objec	Topenes	Export Data	testserver.com	!btA9LSRUgsttRH				
🖽 ≡ PolyBase		Copy Database	estserver.com	EZg4[id)NWVE=D;				
🗄 🛋 Always On High	n Availability	Manage Database Encryption	\dbuse	r (64) TestDatabase	00:00:	10 1	0 row	s
Ready		Database Upgrade						Å

- 2. On the Introduction screen, select Next.
- 3. On the **Column Selection** screen, for **Encryption Type** select **Plaintext**. Then select **Next**.

钮 Always Encrypted	- 🗆 X
Column Selection	
Introduction	🔞 Help
Column Selection Master Key Configuration Run Settings Summary	Search column name Apply one key to all checked columns: MyCKE
Results	Encryption Type 0 Encryption Key 0
	Name State Encryption Type Encryption Key dbo.TestTable - - - LastName - - - LastName - - - Bassword Plaintext MyCKE -
	Show affected columns only
	< Previous Next > Cancel

- 4. On the Master Key Configuration screen, select Next.
- 5. On the Run Settings screen, select Proceed to finish now. Then select Next.
- 6. On the **Summary** screen, verify the configuration choices. Then select **Finish**.
- 7. Present the OCS, select the HSM, and enter the passphrase.
- 8. Check that **Passed** appears in the **Details** column of the **Results** screen.

钮 Always Encrypted			_		\times
Results					
Introduction				🕜 Н	elp
Column Selection					
Master Key Configuration					
Run Settings					
Summary					
Results	Summany:				
	Task			Details	
	Performing encryption operations			Passed	
	Always Encorpted Wizard Log Report				
	ranays encrypted wizard cog hepott				
		< Previous Nex	:>	Close	



The column has been decrypted in the SQL server. To view the plain text data stored SQL server, reconnect to the server with Always Encrypted disabled, see [encrypt-decrypt-column-with-ssms:::view-encrypted-column].

9. Select Close.

Chapter 7. Encrypt or decrypt a column with PowerShell

To encrypt or decrypt a column with PowerShell:

- Encrypt a column
- Remove column encryption

7.1. Encrypt a column

To encrypt a column:

- 1. Log in to the client using the <domain>\dbuser account.
- 2. Launch PowerShell on the client computer and run the

Encrypt_Column_Named_Password.ps1 script (below).

```
# Import the SqlServer module.
Import-Module SqlServer
# Set up connection and database SMO objects
$sqlConnectionString = "Data Source=MS-SQL-AE-Srv.interop.com; Initial Catalog=TestDatabase; Integrated
Security=True; MultipleActiveResultSets=False; Connect Timeout=30; Encrypt=True;
TrustServerCertificate=True; Packet Size=4096; Application Name=`"Microsoft SQL Server Management Studio`""
$smoDatabase = Get-SqlDatabase -ConnectionString $sqlConnectionString
# If your encryption changes involve keys in Azure Key Vault, uncomment one of the lines below in order to
authenticate:
# * Prompt for a username and password:
#Add-SqlAzureAuthenticationContext -Interactive
# * Enter a Client ID, Secret, and Tenant ID:
#Add-SqlAzureAuthenticationContext -ClientID '<Client ID>' -Secret '<Secret>' -Tenant '<Tenant ID>'
# Change encryption schema
$encryptionChanges = @()
# Add changes for table [dbo].[TestTable]
$encryptionChanges += New-SqlColumnEncryptionSettings -ColumnName dbo.TestTable.Password -EncryptionType
Randomized -EncryptionKey "MyCEK"
Set-SqlColumnEncryption -ColumnEncryptionSettings $encryptionChanges -InputObject $smoDatabase
```

The command line is:

> PowerShell -ExecutionPolicy Bypass -File Encrypt_Column_Named_Password.ps1

- 3. Present the OCS, select the HSM, and enter the passphrase.
- Launch Microsoft SQL Server Management Studio. Do as indicated in encryptdecrypt-column-with-powershell:::encrypt-decrypt-column-with-ssms.pdf to verify the column has been encrypted.

7.2. Remove column encryption

To remove column encryption:

1. Launch PowerShell on the client computer and run the

Decrypt_Column_Named_Password.ps1 script (below).

```
# Import the SqlServer module.
Import-Module SqlServer
# Set up connection and database SMO objects
$sqlConnectionString = "Data Source=MS-SQL-AE-Srv.interop.com; Initial Catalog=TestDatabase; Integrated
Security=True; MultipleActiveResultSets=False; Connect Timeout=30; Encrypt=True;
TrustServerCertificate=True; Packet Size=4096; Application Name=`"Microsoft SQL Server Management Studio`""
$smoDatabase = Get-SqlDatabase -ConnectionString $sqlConnectionString
# If your encryption changes involve keys in Azure Key Vault, uncomment one of the lines below in order to
authenticate:
# * Prompt for a username and password:
#Add-SqlAzureAuthenticationContext -Interactive
# * Enter a Client ID, Secret, and Tenant ID:
#Add-SqlAzureAuthenticationContext -ClientID '<Client ID>' -Secret '<Secret>' -Tenant '<Tenant ID>'
# Change encryption schema
$encryptionChanges = @()
# Add changes for table [dbo].[TestTable]
$encryptionChanges += New-SqlColumnEncryptionSettings -ColumnName dbo.TestTable.Password -EncryptionType
Plaintext
Set-SqlColumnEncryption -ColumnEncryptionSettings $encryptionChanges -InputObject $smoDatabase
```

The command line is:

> PowerShell -ExecutionPolicy Bypass -File Decrypt_Column_Named_Password.ps1

- 2. Present the OCS, select the HSM, and enter the passphrase.
- 3. Launch **Microsoft SQL Server Management Studio**. Do as indicated in encryptdecrypt-column-with-powershell:::encrypt-decrypt-column-with-ssms.pdf to verify the column has been encrypted.

Chapter 8. Test access to Always Encrypted keys by another user

To test access to Always Encrypted keys by another user:

- 1. Log in to the client using the <domain>\dbuser2 account.
- 2. Launch Microsoft SQL Server Management Studio.
- 3. Connect to the database on the remote SQL server, enabling Always Encrypted.
- 4. Present the OCS, select the HSM, and enter the passphrase.
- 5. Perform operations on the TestDatabase, which is possible since <domain>\dbuser2 has access to the same MyCMK and MyCEK keys created by <domain>\dbuser.

Chapter 9. Supported PowerShell SqlServer cmdlets

PowerShell cmdlet	Description
Add-SqlColumnEncryptionKeyValue	Adds a new encrypted value for an existing column encryption key object in the database.
Complete-SqlColumnMasterKeyRotation	Completes the rotation of a column master key.
Get-SqlColumnEncryptionKey	Returns all column encryption key objects defined in the database, or returns one column encryption key object with the specified name.
Get-SqlColumnMasterKey	Returns the column master key objects defined in the database, or returns one column master key object with the specified name.
Invoke-SqlColumnMasterKeyRotation	Initiates the rotation of a column master key.
New- SqlAzureKeyVaultColumnMasterKeySettings	Creates a SqlColumnMasterKeySettings object describing an asymmetric key stored in Azure Key Vault.
New-SqlCngColumnMasterKeySettings	Creates a SqlColumnMasterKeySettings object describing an asymmetric key stored in a key store supporting the Cryptography Next Generation (CNG) API.
New-SqlColumnEncryptionKey	Creates a new column encryption key object in the database.
New-SqlColumnEncryptionKeyEncryptedValue	Produces an encrypted value of a column encryption key.
New-SqlColumnEncryptionSettings	Creates a new SqlColumnEncryptionSettings object that encapsulates information about a single column's encryption, including CEK and encryption type.

PowerShell cmdlet	Description
New-SqlColumnMasterKey	Creates a new column master key object in the database.
New-SqlCspColumnMasterKeySettings	Creates a SqlColumnMasterKeySettings object describing an asymmetric key stored in a key store with a Cryptography Service Provider (CSP) supporting Cryptography API (CAPI).
Remove-SqlColumnEncryptionKey	Removes the column encryption key object from the database.
Remove-SqlColumnEncryptionKeyValue	Removes an encrypted value from an existing column encryption key object in the database.
Remove-SqlColumnMasterKey	Removes the column master key object from the database.
Set-SqlColumnEncryption	Encrypts, decrypts or re-encrypts specified columns in the database.

The full list of cmdlets and additions to the SqlServer module can be found in the Microsoft Online Documentation.

- 10.1. nShield Connect
- 10.2. nShield as a Service
- 10.3. Entrust products
- 10.4. nShield product documentation