

Member of Microsoft Intelligent Security Association

Microsoft Security

Microsoft SQL Server 2019 Always Encrypted

nShield[®] HSM Integration Guide

2023-12-05

© 2023 Entrust Corporation. All rights reserved.

Table of Contents

1. Introduction
1.1. Product configurations
1.2. Supported nShield hardware and software versions
1.3. Role separation
1.4. Multiple Windows user accounts on a single on-premises client server 3
1.5. Multiple on-premises client servers
1.6. Always Encrypted and TDE
2. Configure computers and accounts
2.1. Join the domain
2.2. Create domain accounts
3. Install and configure on-premises client
3.1. Select the protection method
3.2. Install the Security World software and create a Security World 5
3.3. Create the OCS or Softcard
3.4. Install and register the CNG provider
3.5. Install and configure SqlServer PowerShell module
3.6. Install the SQL Server Management Studio
3.7. Allow Active Directory user to remote login
4. Install and configure SQL server
4.1. Install the SQL database engine
4.2. Create the SQL logins
5. Generate the encryption keys
5.1. Generate the Always Encrypted Column Master Key (CMK)
5.2. Generate My Column Master Key (MyCMK) and My Column Encryption
Key (MyCEK) with SSMS
5.3. Generate MyCMK and MyCEK with PowerShell
6. Encrypt or decrypt a column with SSMS
6.1. Encrypt a column
6.2. View an encrypted column
6.3. Remove column encryption
7. Encrypt or decrypt a column with PowerShell
7.1. Encrypt a column
7.2. Remove column encryption
8. Test access to Always Encrypted keys by another user
9. Supported PowerShell SqlServer cmdlets

Chapter 1. Introduction

Always Encrypted is a feature in Windows SQL Server 2019 designed to protect sensitive data both at rest and in flight between an on-premises 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 on-premises client application server. This effectively mitigates manin-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 on-premises client application server.

1.1. Product configurations

Entrust successfully tested nShield HSM integration with Windows SQL Server 2019 and the Always Encrypted feature in the following configurations:

1.1.1. Remote server

Product	Version
SQL Server	Microsoft SQL Server 2019
Base OS	Windows Server 2019 Datacenter

1.1.2. On-premises 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

Product	Security World Software	Firmware	Netimage	OCS	Softcard	Module
Connect XC	12.80.4	12.72.1 (FIPS Certified)	12.80.5	\checkmark	\checkmark	\checkmark
nShield 5c	13.2.2	13.2.2 (FIPS Pending)	13.2.2	\checkmark	\checkmark	\checkmark
nSaaS	12.80.4	12.72.1 (FIPS Certified)	12.80.5	\checkmark	\checkmark	\checkmark

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

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 onpremises client server

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

1.5. Multiple on-premises client servers

Each on-premise 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 on-premises 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 onpremises client

This installation must be performed on the on-premises 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
mode
                    operational
                    12.80.4
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.

- 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 2: blank card
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 non-persistent 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.

 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 1 testOCS	Keys (recov) 0 (0)	Sharing 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.



3. 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 loadsharing 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.

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

t Module Sta Ensure modul	tes es are in the correct s	tate before you proceed.	EN
The following	modules are available	e in your system:	
Module ID	Mode	State	
1	operational	usable	
2	operational	foreign	
At least one n	nodule is usable in the	current world. Click Next to	continue with this world.
uninitialized n	Shield modules.	I state to enable you to rest	ore your security world to
Refer to the u state. If you n	ser guide for details o eed to power down y ard on boot up to cor	f how to put your nShield m our computer, select the tick ntinue the installation.	odule in the initialization kbox below and then

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

nShield CNG Providers Configuration Wizard	\times
Key Protection Setup Set up the private key-protection method.	ENTRUST
Select the default method that will be used to protect private keys generated by the CNG Key Storage Provider.	
If softcard or OCS protection is selected, the choice will be offered on the next page whether to use an existing token or create a new one.	
 Module protection (requires no extra cards but is less secure). Softcard protection (unavailable in HSM Pool Mode). Operator Card Set protection (unavailable in HSM Pool Mode). Allow any protection method to be selected in the GIII when generating 	
< Back Next > C	ancel

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

nShield CNG Providers Configuration	n Wizard		×
Token for Key Protection Select the token that will be used	to protect new keys, or	create a new token.	ENTRUST
Current Operator Card Sets: testOCS	Operator Card Set To Name: Token hash: Sharing parameters: Timeout: Currently protecting	oken Information: test0CS 0xa165a26f 1 of 1, Non-persistent None : none	
Create a new Operator Card Set	t name		
Number of cards required (K)	: To	tal number of cards (N):	
Card set has a time Persistent	out Card set time Usable remotely	eout:se	conds
	< Back	Next >	Cancel

8. Verify the provider with the following commands:

>certutil -<mark>csplist</mark> | 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 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



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.

```
PS C:\Users\Administrator.EXAMPLE> Get-Module -list -Name SqlServer
Directory: C:\Program Files\WindowsPowerShell\Modules
ModuleType Version Name ExportedCommands
------
Script 21.1.18256 SqlServer {Add-RoleMember, Add-SqlAvailabilityDatabase, Add-SqlAvail...}
```

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:
 - domain>\dbuser
 - domain>\dbuser2.

	Advanced Re	emote	
Remote Desktop Users			?)
The users listed below ca the Administrators group	n connect to this can connect eve	computer, and a n if they are not li	ny members o sted.
\dbuser			
aduser2			
Administrator a	already has acce	SS.	
Add Rer	nove		
To create new user accou	unts or add users	s to other groups,	go to Control
Taner and open oser Acc	<u>Journa</u> .		
			Ornerl
		OK	Cancer

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.

File Action View Help												
🗢 🏟 🙍 📰 🗟 🚺												
P Windows Defender Firewall with	Inbound Rules											
Cuthound 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
> S Monitoring	🔮 Firefox (C:\Program Files\Mozilla Firefox)		Private	Yes	Allow	No	C:\Progr	Any	Any	TCP	Any	Any
~ <u> </u>	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								
🍵 🗢 🗸 🐮 🔹 🐂 🛀 🔛 🚰 🗍 New Query 🗯 🖓 🖓 🕯		¥ 🗅 ည	9-	 ~ ⊠ ~ . 		- 😡	۵	⊳
🕴 🚏 💜 TestDatabase 🔹 🕨 Execute 🔳 🖌 🛱 🗐		80 82 🗊	8 8 1	🖸 🗏 🖆 🚣 🛛 🍋	÷			
	SOLO)uerv1 sal - N	45 Admin	istrator (57)) 👍 🗙				-
	- Cart	/*****	Script f	or SelectTopNRows com	mand from SSMS	*****/		÷
Connect • • • • • • • • •		SELECT T	OP (1000) [FirstName]				
🗆 🖬 MS-SQL-AE-Srvcom (SQL Server 15.0. \Ar 🔨		٦,	LastName]				
🖃 💻 Databases		.[Email]					
🗄 💻 System Databases			Password]				
🗄 📁 Database Snapshots		FROM [TestData	base].[dbo].[TestTable	e]			
🗆 🖶 TestDatabase								
🗉 💻 Database Diagrams								
🖂 💻 Tables								
🕀 💻 System Tables								
⊞ = FileTables								-
🕀 🖷 External Tables	100 9	6 - 4						
🗄 🖷 Graph Tables	■R	esults ell Mos	ance					
🖽 🎟 dbo.TestTable		FirstName	LastName	Email	Password			
🗄 🖷 Views	1	Jack	Shepard	jack.shepard@testserver.com	%#[BgT.z4B& UM5			<u>^</u>
🗄 🖷 External Resources	2	John	Locke	john.locke@testserver.com	v@2Mxr;XYcYsIPw			
🗄 🖷 Synonyms	3	Kate	Austin	kate.austin@testserver.com	I!8wbgcg85#I[
🗄 🖷 Programmability	4	James	Ford	james.ford@testserver.com	J5YPbd59w\$5siuk			
🗉 💻 Service Broker	5	Ben	Linus	ben.linus@testserver.com	MY1=g=&gm{.UATC			
🗄 💻 Storage	6	Desmond	Hume	desmon.hume@testserver.com	aPoTEp)h;TfNWT1			
E Security	7	Daniel	Faraday	daniel.faraday@testserver.com	9MPDzVhXYJSJQ%%			
H Security	8	Sayid	Jarrah	sayid.jarran@testserver.com	GTONIXIJ[H{m9w}			
🗄 🖷 Server Objects 🗸 🗸	9	lacob	Apert	incinaru.alpert@testserver.com	EZa/IId/N/W/E=D			~
< · · · · · · · · · · · · · · · · · · ·	0 Q	Je 🔒 MS-	-SOL-AE-SI	vcom (\A	dministrator Test	Database 00:0	0:00 1	0 rows
					Note the second	50.0		
🗖 Ready Ln 6 Col 31	Ch	31		NS				4

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 of TestDatabase.

Login Properties -	\dbuser		-		\times			
Select a page General	IT Script ▼							
Server Roles User Mapping Serurables	Users mapped to this login:							
 Status 	Map Database User master model model		Default Schema					
L	TestDatabase	dbuser	dbo					
Connection Server:	Guest account enabled for mas	ter						
MS-SQL-AE-SRV Connection: Administrator	Database role membership for: mas	ster						
View connection properties	db_backupoperator db_backupoperator db_datareader db_datawriter db_ddladmin db_denydatareader db_denydatareader							
Progress Ready	db_owner db_securityadmin							
			ОК	Can	icel			

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 on-premises client using the <domain>\Administrator, 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.provider = $cngProvider
$cngKeyParams.provider = $cngProvider
$cngKeyParams.keyCreationOptions =
[System.Security.Cryptography.CngKeyCreationOptions]::OverwriteExistingKey
$keySizeProperty = New-Object System.Security.Cryptography.CngProperty("Length",
[System.BitConverter]::GetBytes($cngKeySize), [System.Security.Cryptography.CngProperty("Length",
[System.BitConverter]::GetBytes($cngKeySize), [System.Security.Cryptography.CngProperty("Length",
[System.BitConverter]::GetBytes($cngKeySize), [System.Security.Cryptography.CngPropertyOptions]::None);
$cngAlgorithm = New-Object System.Security.Cryptography.CngAlgorithm($cngAlgorithmName)
$cngKey = [System.Security.Cryptography.CngKey]::Create($cngAlgorithm, $cngKeyName, $cngKeyParams)
```

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 Cancel	

- 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 - Create key	
	Select token to protect the key with.	
	Current Operator Card Sets: Derator Card Set Token Information: Name: testOCS Token hash: 0xa165a26f Sharing parameters: 1 of 1, Non-persistent Timeout: None Currently protecting: none	
	<u>N</u> ext Cancel	

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

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

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.

				\times
nCipher K	ey Storage Pr	ovider		
ard readi	ng comple	ete.		
Module	Slot	Content	Status	
	5		complete	
	3		complete	
	2		complete	
	Ŭ.		compiete	
				Finish
	nCipher Ki	nCipher Key Storage Pr ard reading completed fodule Slot 5 4 3 2 0	nCipher Key Storage Provider ard reading complete. 10dule Slot Content 5 4 3 2 0	nCipher Key Storage Provider ard reading complete. 10dule Slot Content Status 5 complete 4 complete 3 complete 2 complete 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 On-Premise 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 other flags hkm hkt hkr	6 Token 0x0 28ee9f7cfceba95992f1f3f31b39c8dba7cfa960 a165a26f929841fe9ff2acdf4bb6141c1f1a2eed none
BlobRecoveryKA format other flags hkm hkt hkr	9 UserKey 0x0 none none 55c38c84103d95278fd54b6b5b3e67d614db8538
BlobPubKA format other flags hkm hkt hkr	5 Module 0x0 c2be99fe1c77f1b75d48e2fd2df8dffc0c969bcb none 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 on-premises 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		>
	SQL Server	
Login Connection Properties Server	Always Encrypted Additional Connection Parameters	
Server type:	Database Engine	~
Server name: Authentication:	MS-SQL-AE-Srvcom Windows Authentication	~
User name:	\dbuser	~
Password:	Remember password	
	Connect Cancel Help Option	ns <<

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

Connect to Server				>
	SQL S	Server		
Login Connection Propertie	Always Encryp	ted Additional Co	nnection Pa	arameters
Type or select the name of t	he database for t	ne connection.		
Connect to database:	<default></default>			~
Network				
Network protocol:	<default></default>			×
Network packet size:	4096 🗘	bytes		
Connection				
Connection time-out:	30 🗘	seconds		
Execution time-out:	0	seconds		
Encrypt connection		2		
✓ Trust server certifica	te			
Use custom color:		Select		
				Reset All
	Connect	Cancel	Help	Options <<

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

⊒ [■] 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.	1
Learn Mo	ore
Connect Cancel Help Options	<

- 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.



- 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:	7			
	nCipher Sec	urity World Key Storage Provider V				
	Name AECMK					-
Connection						
Server: MS-SQL-AE-SRV						
Connection: \dbuser						
View connection properties						
Progress						
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**.



- 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 Key	y		-		\times
Select a page	🗊 Script 🔻 😯 Help				
	Name:	МуСЕК			
	Column master key:	МуСМК		Refresh	
	Column encryption k encryption keys. This	eys protect your data, and column master keys prote lets you manage fewer keys.	ct your colu	ımn	
	To create a new colu	ımn master key, use the "New Column Master Key" p	age.		
Connection					
Server:					
MS-SQL-AE-SRV					
Connection: \dbuser					
View connection properties					
Progress					
Ready					
			OK	Cano	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**.

Choose modules y	ou wish to load the key onto.	
Excluded modules:	Included modules:	
Module #2	Add Module #1 Remove	
	Add all	
	Remove all	

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	I

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

nCipher Key Storage Provider

Card re	eading	compl	ete.
Module	Slot	Content	Status
1	5		complete
1	4		complete
1	3		complete
1	2		complete
1	0		complete

Finish	

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

 \times



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:



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

爾 Always Encrypted	– 🗆 X
Introduction	@ Help
Column Selection	
Master Key Configuration	
Run Settings	Always Encrypted is designed to protect sensitive information - such as credit card numbers - stored in SQL Server databases. It enables clients to encrypt data inside client applications and never reveal the
Summary	encryption keys to SQL Server.
Results	
	, <u>n</u> ,
	[]
	oo nor snow mis page again.
	< Previous Next > Cancel

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

			- 0	>
Column Selection				
Introduction			(🕢 Help
Column Selection				
Master Key Configuration	Search column name			
Run Settings	Apply one key to all checked columns:		MyCEK	\sim
oummary		Encryption Type	① Encryption K	ey 🛈
lesults	Name State	Encryption Type	Encryption Key	
	dbo. Iest lable dbo. Iest lable ListName EstName EstName			
	Password 🥒	Randomized	 MyCEK 	-
	Show affected columns only			

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

団 Always Encrypted		-		×
Master Key Configu	uration			
Introduction			@ F	lelp
Column Selection	No additional configuration is necessary because you are using existing keys.			
Master Key Configuration				
Summary				
Results				
	< Previous Next >		Cance	I

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.
	Select how you would like to proceed O Generate PowerShell script to run later O Proceed to finish now
	< Previous Next > Cancel

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

钮 Always Encrypted		_		×
Summary				
Introduction Column Selection			🔞 H	lelp
Master Key Configuration	Verify the choices made in this wizard.			
Run Settings	Click Finish to perform the operations with the following settings:			
Summary Results	Source database sating: Source database name: TestDatabase Encrypt column Pasword Encryption key name: MyCEK Encryption key name: MyCEK Encryption type: Randomized	1	Cancer	al

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

碅 Always Encrypted			-		×
Results					
Introduction				🕜 H	dp
Column Selection					
Master Key Configuration					
Run Settings					
Summary	Summary:				
Results	Task		Details		
	Performing encryption operations		Passed		
					T
	Alumin Francisco Millional Law David				
	Anways encrypted wizard Log Report				
		< Previous Next	>	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 on-premises client. This is because **Always Encrypted** is performing the decryption at the on-premises 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	×
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.	1
Learn Mr	ore
Connect Cancel Help Options	:<

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.

🔀 SQLQuery2.sql - MS-SQL-AE-Srvcom.TestDatabase (\dbuser	(68)) - 1	Microsoft SC	L Server M	anagement Studio	Quick Lau	inch (Ctrl+Q)	٩	-	×
File Edit View Project Tools Window Help	ile Edit View Project Tools Window Help								
💿 🗸 💿 🛛 📸 👻 🎦 👻 🎬 🚰 💭 New Query 🔎 🔊 🖓 🖓 🖓		â 🤊	- 9 - 1	SI - 5		•	2	۵	▶ • -
8 ♥ 1 TestDatabase → ► Execute ■ ✓ 80 @				2월 고준 글드 30	_				
	SOLO)uerv2 sal - I	M	dbuser (68)) 📲 X		v1sal-M ∖v	dhuser (55))	
	Jaco	/*****	Script f	or SelectTopNRo	vs com	mand from SSMS	*****	*/	÷
Connect - T T V C	E E	SELECT T	OP (1000) [FirstName]				·	-
SSQL-AE-Srvcom (SQL Server 15.0. \dbuser)		[LastName]					
🖃 🗯 Databases		[Email]						
🗄 💻 System Databases			Password]					
🗄 💻 Database Snapshots		FROM [TestData	base].[dbo].[Te	stTable	e]			
TestDatabase									
🗄 💻 Database Diagrams									
🗆 📁 Tables									
🗄 🖷 System Tables									
🗄 🗯 FileTables									
🗄 💻 External Tables									_
🗄 💻 Graph Tables	100.0/								- h
⊞ III dbo.TestTable	100 %								r
🕀 💻 Views	III RE	esuits de Me	ssages	F 3					_
🗄 💻 External Resources	1	FirstName	LastName	Email		Password 0x014DDAAAA3696A	DZOODD	7001000	CA 751
III = Synonyms	1	John	Locko	jack.sneparu@testsen	com	0x014DBAAA2000A	B/00BD	70D IE20	421D
III = Programmability	2	Kate	Austin	kate austin@testserve	r com	0x0191EBC7A673E	D761A57	7E6E832	5ABA
III = Service Broker	4	James	Ford	james.ford@testserver	com	0x01E424BA5D5CE	229D74F	215DB52	27062
🗄 🖷 Storage	5	Ben	Linus	ben.linus@testserver.c	:om	0x0186A978A760BF	B915E1	39C4986/	A46A2
= Security	6	Desmond	Hume	desmon.hume@testse	rver.com	0x014F64DD625EA	921CC6E	58D96F6	D882
	7	Daniel	Faraday	daniel.faraday@testse	rver.com	0x0166C0781A0951	56532FF	4C584D0	69DB
I Roles	8	Sayid	Jarrah	sayid.jarrah@testserve	r.com	0x010CB86E859D4I	D68CD4A	9C1D16	371AF
H Schemas	9	Richard	Alpert	richard.alpert@testsen	ver.com	0x017D746C6F36FA	A346FC	89D3C1E	78A6
Symmetric Keys	10	Jacob	Smith	iacob.smith@testserve	r.com	0x0124483B58C627	AE21572	C947986	AA1D Y
Certificates		0.00					00.0		
- Certificates	Q	MS-SQ	-AE-Srv.	.com (\dbuse	er (68) IestDatabas	e 00:00	0:00 10	rows
C Ready									

- 3. Reconnect to the SQL server, but with the **Enable Always Encrypted** checked.
- 4. Present the OCS, select the HSM, and enter the passphrase.
- 5. 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-Srv	Management Studio Quick Lau	nch (Ctrl+Q)	- ۹	□ ×
File Edit View Query Project Tools Window Help				
💿 🔹 💿 🖹 🔹 'n - 😩 💾 🚰 💭 New Query 🔎 🔊 🖓 🎧 🎧 🎧 🛔 日 白 🛛 🏷 - 🤍 -	8 - 5	-	N P :	Ê ⊳
🛛 🕆 💜 TestDatabase 🔹 🗸 🕨 🖬 🖬 🕼 3	2 - · · · · · · · · · · · · · · · · · ·			
Object Explorer	\dbuser (64)) ≄ × SQLQuer	y2.sql - M \dbı	user (68))	÷
Connect → ♥ ×♥ ■ ▼ C →	for SelectTopNRows comm 0) [FirstName]	nand from SSMS *	*****/	÷
🗆 🗟 MS-SQL-AE-Srvcom (SQL Server 15.0. \dbuser) ^ , [LastNam	e]			- T
= 🖬 Databases , [Email]	-			
🗄 🖷 System Databases , [Passwor	d]			
⊞ = Database Snapshots FROM [TestDat	abase].[dbo].[TestTable	2]		- 11
🗆 🗎 TestDatabase				- 11
🗄 📁 Database Diagrams				
🖃 💻 Tables				
🗄 💻 System Tables				
🗄 💻 FileTables				
🗄 🗯 External Tables				_
🗄 🖷 Graph Tables				
⊞ dbo.TestTable				· ·
H Views	Prove 1	Provide the second seco	1	
External Resources	izek shonord@testsoner.com	Password 9/#fpatia4P8_LIME		
E Synonyms 2 John Locke	jack.sheparu@testserver.com	v@2Myr:XYcYsIPw		
Programmability A Kate Austin	kate austin@testserver.com	l'8wbaca85 #lf		
Service Broker 4 James Ford	james.ford@testserver.com	J5YPbd59w\$5siuk		
🗄 🖷 Storage 5 Ben Linus	ben.linus@testserver.com	MY1=g=&gm{.UATC		
Becurity 6 Desmond Hume	desmon.hume@testserver.com	aPoTEp)h;TfNWT1		
B Security 7 Daniel Faraday	daniel.faraday@testserver.com	9MPDzVhXY]S]Q%%		
B Sayid Jarrah	sayid.jarrah@testserver.com	Gfonlxi][H{m9w}		
9 Richard Alpert	richard.alpert@testserver.com	!btA9LSRUgsttRH		
B PolyBase	jacob.smitn@testserver.com	EZg4[Id)NWvE=D;		
E = Always On High Availability	.com (\dbuse	er (64) TestDatabase	00:00:10	10 rows
Ready In 1 Col 1 Ch 1 INS		. ,		

6.3. Remove column encryption

 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 Stu	dio Quick Lau	nch (Ctrl+Q)	P	-		×
File Edit View	Project Tools Window Help							
© ▼ © 🏠 ▼ '	🛅 👻 🎬 🚰 🛛 💭 New Query 🖉	Detach		•	111	۵ م	Σ	* ÷
🕴 🕴 🕅 TestDatab	ase 👻 🕨 Execute 🔳	Take Offline	*@ 🚽					
Object Surglasses		Bring Online		vΩeal M i ∖db	uncor 16	9))		=
Object Explorer	- • •	Stretch +	nNRows com	and from SSMS	*****	*/		-
Connect 🕶 🌹 🍟 🗏	Y C -*	Encrypt Columns	ne]			· ·		
B MS-SQL-AE-Srv.	.com (SQL Server 15.0.	Data Discovery and Classification						
Databases		Vulnerability Assessment						
E = System Database Spa	inshots	Shrink +].[TestTable	1				
🗆 🛢 TestDatab		Back Up		•				
🗄 💻 Databas	New Database	Restore						
🖯 💻 Tables	New Query	Mirror						- 11
🗄 📫 Syste	Script Database as	Launch Database Mirroring Monitor						
🕀 🖷 FileTa	Tasks •	Ship Transaction Logs						
Extern	Policies •	Generate Scripts						Ŧ
⊞ ≡ dbo.1	Facets	Generate In-Memory OLTP Migration Checklists					•	
🗄 📁 Views	Start PowerShell	Extract Data-tier Application		Descended				
🗄 💻 External	Azure Data Studio	Deploy Database to Microsoft Azure SOL Database	testserver.com	%#fBgT.z4B& UM5				
🗄 💻 Synonyi	Azure SQL Managed Instance link 🔸	Export Data-tier Application	stserver.com	v@2Mxr,XYcYsIPw				
🗄 📫 Progran	Reports •	Register as Data-tier Application	stserver.com	I!8wbgcg85#I[
H = Service	Rename	Upgrade Data-tier Application	stserver.com	J5YPbd59w\$5siuk MY1=g=&gp(11ATC				
III = Storage	Delete	Delete Data-tier Application	@testserver.com	aPoTEp)h;TfNWT1				
E = Security	Refresh	Import Flat File	2testserver.com	9MPDzVhXYJSJQ%%				
🗄 🖷 Server Objec	Properties	Import Data	estserver.com	Gfontxi][H{m9w}				
🗄 ≡ Replication		Export Data	estserver.com	EZ04[Id)NWvE=D				
🖽 💻 PolyBase		Copy Database						_
🗄 💻 Always On High	Availability	Manage Database Encryption	\dbuse	r (64) TestDatabase	00:00	:10 10	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	-	<
Column Selection		
Introduction	🕑 Help	
Master Key Configuration Run Settings Summary	Search column name Apply one key to all checked columns: MyCKE	
Kesuits	Encryption Type 0 Encryption Key 0	
	Name State Encryption Type Encryption Key dbo.TestTable FirstName LastName LastName Ernail Paintext Password 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.

🔁 Always Encrypted		-		×
Results				
Introduction			🕜 Help)
Column Selection				
Master Key Configuration				
Run Settings				
Summary				
Results	Summane			
	Tack		otaile	1
	Performing encryption operations	Pa	assed	
	Always Encrypted Wizard Log Report			
	< Previous Next >		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 [encryptdecrypt-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 on-premises client using the <domain>\dbuser account.
- Launch PowerShell on the on-premises 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 encrypt-decrypt-column-with-powershell:::encrypt-decrypt-column-withssms.pdf to verify the column has been encrypted.

7.2. Remove column encryption

To remove column encryption:

 Launch PowerShell on the on-premises 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:



- 2. Present the OCS, select the HSM, and enter the passphrase.
- 3. Launch **Microsoft SQL Server Management Studio**. Do as indicated in encrypt-decrypt-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 on-premises 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.
- 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.

PowerShell cmdlet	Description
New-SqlColumnEncryptionSettings	Creates a new SqlColumnEncryptionSettings object that encapsulates information about a single column's encryption, including CEK and encryption type.
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.