Activate Always Encrypted Capability

Last Modified on 03/31/2020 9:03 am EDT

Overview

To ensure that sensitive data is protected, Cora SeQuence supports various operations such as symmetric and asymmetric encryption and decryption, and hashing operations within the application. Cora SeQuence offers encryption features such as Public Key Infrastructure (PKI) support and signed workflow packages, and supports various key store locations, such as internal, certification store, and Microsoft Azure key-vault. You can also use machine key for xsrf token encryption.

Following are the main features supported for encryption in Cora SeQuence:

- Symmetric encryption and decryption
- Asymmetric encryption and decryption
- Hashing
- Signature validation
- PKI support
- Consider using Azure Key-Vault
- Consider Key Stores locations such as Internal, Cert Store and Key Vault
- Sign Packages
- Consider use machine key for xsrf token encryption

Enable always encrypted capability

To activate the always encrypted capability in Cora SeQuence, enable the encrypted connections to SQL Server Database Engine.

Prerequisites

- Rijndael Salt
- Rijndael Key
- SHA256 Hash Salt
- Microsoft SQL Server Management Studio (SSMS) 2016 SP1 or higher

Certification creation (using SSMS)

1. Generate column encryption.
2. Open SSMS.
3. Click on the relevant DB.
4. From the folders expand the Security folder.
5. Right click the Always Encrypted Keys folder and select ‘New Column Master Key… ’.
6. On the ‘New Column Master Key’ window, enter ‘Always Encrypted Certificate’ in the Names textbox.
7. Click ‘Generate Certificate’ to generate a new certificate and it's thumbprint.
8. On your PC, open the Console (on Run, enter MMC and press Enter).
9. Click on the File menu and select ‘Add/Remove Snap-in….’.
10. On the Add or Remove Snap-ins window, from the list of Available snap-ins, select the certificates to be added.
11. Click Add >.
12. Choose “Computer account” and click finish <>

13. Click OK. The Certificates are added to the Selected snap-ins list.

14. On Console window, open the certificates folder to view the recently added certificates.

    To apply the certificate on all servers connected to the DB, export and add the certificate to these servers.

15. Right click the certificate and select 'All Tasks',
16. Click 'Export...'.
17. Save the certificate locally.
18. Copy the Certificate thumbprint created in steps above.
19. Double click the certificate.
20. On the Certificate window, click the Details tab.
21. Select and copy the Value for Thumbprint.

22. Open the first DB script file (000. CREATE MASTER KEY AND ENCRYPTION KEY.sql).

Database Initialization

1. Open the DB script file (as mentioned in step 22 above) and review the code.
2. Modify the first command USE ??? to add the relevant DB on which the Always Encrypted Certificate is to be enabled.

   > **NOTE**:
   > The question marks in any command suggests it requires modification.

3. Modify the second command to add the copied Thumbprint value.
4. Click Execute or press F5 to execute the script. The output 'The command(s) completed successfully' is received.

5. Before closing the table with the always on feature, enter the values into the table.
6. Insert Customer Keys and Salt Values using the DB script.

```
INSERT INTO [sec].[tblSecrets]
           ([fldKey]
           ,[fldValue]
           ,[fldActive])
     VALUES
           (
           ,
           ,)
GO
```

Column encryption key initialization

The column encryption key initialization is done by executing the PowerShell script (Encrypt_Column.ps1). To initialize, follow the steps below:

1. Open the PowerShell script.
2. Change the [string]$DataSource parameter value with the server name where the DB is located.
3. Change the [string]$InitialCatalog parameter value with the DB name.
4. After those 2 parameters have been provided we can continue and execute the script.
5. Click Execute or press F5 to execute the script. The output 'The command(s) completed successfully'

is received.
6. In the DB right click the sec.tblSecrets table to see the table creation code.
7. Execute the generate script code to check if there is encryption on the fldValue column.

Cryptographic services and providers activation

Import certification

1. Open local machine Certificates snap-in using MMC.
2. Right click and open Personal certifications folder.
3. Select All tasks > Import... to open Certificate Import Wizard.
4. Make sure the Local Machine is selected for store location.
5. Click Next to continue.
6. Click Browse... and select file to import certificate from. Make sure the certificate file type is correctly selected, otherwise some file types will not be visible (for example .ipx).
7. Click Next to import.

Activate cryptographic services

In the configuration, add the following entry under **sequence.engine\services**.

Set up Rijndael cryptographic provider

In the configuration, add the following entries:

- Under the section group of **sequence.engine**.
- Under the **sequence.engine\cryptoProvider**

Setup SQL secrets providers

Under **sequence.engine\persistence\providers**.

Usage (code)

1. Obtain the CryptoServices (engine.GetService).
2. Encryption

```
byte[] CryptoServices.CryptoProvider.Encrypt(byte[] data)
```

3. Decryption

```
byte[] CryptoServices.CryptoProvider.Decrypt(byte[] cipher)
```