Upgrade Cryptographic Service Implementation

Last Modified on 08/24/2023 3:21 am EDT

V10.4

Overview

Cora Orchestration supports store and fetch of encrypted values like user credentials, connection strings, database credentials, API keys, OAuth tokens, and other secrets for the configuration files at runtime. Cora Orchestration versions earlier than V10.4 uses the Rijndael algorithm for encryptions.
For details, see this article and this link.

Since the Rijndael is obsolete from .NET6 onwards, we have upgraded to use AES (Advanced Encryption Standard) algorithm for enhanced security. Cora Orchestration now uses AES ECB mode for encryptions.
For details, see this link.

With this upgrade of cryptographic algorithm, Cora Orchestration by default comes with a new AESCryptoProvider that uses the following IDs by default:

```
KeyId = "sequence:cryptography:aes:key";
```

```
SaltId = "sequence:cryptography:aes:salt";
```

You can configure these IDs, if required.

The Rijndael crypto provider is supported with optional configuration, only for existing customers who have encrypted their data using Rijndael algorithm. For any new encryption AESCryptoProvider is recommended.

Procedure

Add to the key vault the following two AES keys, and set them in the configuration.

For AES crypto provider:

- sequence:cryptography:aes:key
- sequence:cryptography:aes:salt

The key for hash provider already exists and remains the same.

- sequence:cryptography:sha256:salt

Backward compatibility

Upgrade from Rijndael to AES is not backward compatible because of the use of ECB encryption mode. To use AES cryptographic algorithm, you must encrypt the data from scratch i.e. re-enter all sensitive data manually for encryption.

If you have encrypted data with Rijndael algorithm and want to keep using the encrypted data, you must use the Rijndael crypto provider, and the following Rijndael crypto keys.

- sequence:cryptography:rijndael:key
- sequence:cryptography:rijndael:salt