IBM WebSphere MQ: Integration Guide

THALES LUNA HSM

Document Information

Document Part Number	007-011561-001
Revision	H
Release Date	17 April 2024

Trademarks, Copyrights, and Third-Party Software

CONTENTS

Overview

This guide is intended to provide instructions for setting up a small test lab that has IBM WebSphere MQ running with Luna HSM to secure the SSL private keys, public keys, and certificates. The guide explains installation and configuration of Luna Client software with IBM WebSphere MQ for storing SSL keys and certificates on Luna HSM.

The integration between Luna HSM and IBM WebSphere MQ uses the IBM Java interface to generate the keys and certificates on Luna HSM. Integration of IBM WebSphere MQ with Luna HSM generates SSL keys. Luna HSM provides security by protecting the private keys within a FIPS 140-2 certified hardware security module.

The benefits of using Luna HSM to generate the SSL keys for IBM WebSphere MQ include the following:

- Secure generation, storage, and protection of the SSL keys on FIPS 140-2 level 3 validated hardware.
- Full life cycle management of the keys.
- HSM audit trail.
- Significant performance improvements by off-loading cryptographic operations from servers.

About IBM WebSphere MQ

IBM WebSphere MQ supports the exchange of information between applications, systems, services and files by sending and receiving message data via messaging queues. This simplifies the creation and maintenance of business applications. Secure communications that use the TLS cryptographic security protocols involve setting up the communication channels and managing the digital certificates that you will use for authentication.

IBM WebSphere MQ is a family of network software products launched by IBM in March 1992. It was previously known as MQ Series. It allows independent and potentially non-concurrent applications on a distributed system to communicate with each other.

Luna HSMs provides key management security for certificates and certificate-based authentication, including import of trusted CA certificates from software based keystore to hardware based keystore, and generation of self-signed certificates and personal certificate requests via the IBM Key Management Utility.

Certified Platforms

This integration is certified on the following platforms.

Luna HSM: Luna HSM appliances are purposefully designed to provide a balance of security, high performance, and usability that makes them an ideal choice for enterprise, financial, and government organizations. Luna HSMs physically and logically secure cryptographic keys and accelerate cryptographic processing. The Luna HSM on premise offerings include the Luna Network HSM, Luna PCIe HSM, and Luna USB HSMs. Luna HSMs are also available for access as an offering from cloud service providers such as IBM cloud HSM and AWS cloud HSM classic

HSM Type	Platforms Tested	IBM WebSphere MQ
Luna HSM	RHEL	IBM WebSphere MQ v9.3.0.15 IBM WebSphere MQ v9.1.0.12
Luna HSM	Windows Server	IBM WebSphere MQ v9.1.0.2 with IT15253 (Patch 9.1.0.2-IBM-MQ-Win64-TF55486)

The Patch 9.1.0.2-IBM-MQ-Win64-TF55486 fixes the HSM Password issue "AMQ9671E: The PKCS #11 token password specified is invalid." in MQ Client when it communicates to HSM using password provided in MQClient.ini file. The APAR for this fix is IT30722 which can be downloaded from IBM ecurep on request.

> **NOTE:** The above patch is not required for IBM MQ v9.1.0.6 or above. If you are using older versions (7.x or 8.x) of IBM WebSphere MQ, please refer the earlier Luna HSM integration guide "IBM_WebSphere_MQ_Integration Guide_RevE"

Prerequisites

Before you proceed with the integration, complete the following tasks:

Configure Luna HSM Device

To configure a Luna HSM device:

1. Ensure that the HSM is set up, initialized, provisioned, and ready for deployment. Refer to the HSM product documentation for help.
2. Create a partition that will be later used for generating SSL keys for IBM MQ.
3. Create and exchange certificate between the Luna Network HSM and Client system. Register client and assign partition to create an NTLS connection. Initialize Crypto Officer and Crypto User roles for the registered partition.
4. Ensure that the partition is successfully registered and configured. The command to see the registered partitions is:

```
C:\Program Files\SafeNet\LunaClient>lunacm.exe

lunacm.exe (64-bit) v10.4.0-417. Copyright (c) 2021 SafeNet. All rights
reserved.

     Available HSMs:

     Slot Id ->              0

     Label ->                TPA01

     Serial Number ->        1312109862201

     Model ->                LunaSA 7.7.1

     Firmware Version ->     7.7.1
```

```
Bootloader Version ->   1.1.2
Configuration ->        Luna User Partition With SO (PW) Key Export
                       With Cloning Mode
Slot Description ->     Net Token Slot
FM HW Status ->         Non-FM
Current Slot Id: 0
```

5. For PED-authenticated HSM, enable partition policies 22 and 23 to allow activation and auto-activation.

> **NOTE:** Refer to Luna HSM documentation for detailed steps on creating NTLS connection, initializing the partitions, and assigning various user roles.

Set up Luna HSM in FIPS Mode

> **NOTE:** This setting is not required for Luna HSM Universal Client. This setting is applicable only for Luna HSM Client 7.x.

Under FIPS 186-3/4, the RSA methods permitted for generating keys are 186-3 with primes and 186-3 with aux primes. This means that RSA PKCS and X9.31 key generation is no longer approved for operation in a FIPS-compliant HSM. If you are using Luna HSM in FIPS mode, you have to make the following change in the configuration file:

```
Misc = {
RSAKeyGenMechRemap = 1;
}
```

The above setting redirects the older calling mechanism to a new approved mechanism when Luna HSM is in FIPS mode.

> **NOTE:** If IBM MQ is using CKM_SHA1_RSA_PKCS mechanism for signing the data with Luna SA f/w 7.x.x, upgrade the firmware to 7.7.0 or above to use CKM_SHA256_RSA_PKCS mechanism which is supported in FIPS mode.

Configure Luna HSM HA (High-Availability)

Please refer to Luna HSM documentation for HA steps and details regarding configuring and setting up two or more HSM appliances on Windows and UNIX systems. You must enable the HAOnly setting in HA for failover to work so that if primary stop functioning for some reason, all calls automatically routed to secondary till primary starts functioning again.

> **NOTE:** This integration is tested in both HA and FIPS mode.

Controlling user access to HSM

> **NOTE:** This section is applicable only for Linux users.

By default, only the root user has access to the HSM. You can specify a set of non-root users that are permitted to access the HSM by adding them to the `hsmusers` group. The client software installation automatically creates the hsmusers group. The hsmusers group is retained when you uninstall the client software, allowing you to upgrade the software while retaining your hsmusers group configuration.

Add a user to hsmusers group

To allow non-root users or applications access to the HSM, assign the users to the hsmusers group. The users you assign to the hsmusers group must exist on the client workstation.

1. Ensure that you have sudo privileges on the client workstation.
2. Add a user to the hsmusers group.

   ```
   # sudo gpasswd --add <username> hsmusers
   ```

   Where <username> is the name of the user you want to add to the hsmusers group.

Removing a user from hsmusers group

3. Ensure that you have sudo privileges on the client workstation.
4. Remove a user from the hsmusers group.

   ```
   # sudo gpasswd -d <username> hsmusers
   ```

   Where <username> is the name of the user you want to remove from the hsmusers group. You must log in again to see the change.

   > **NOTE:** The user you delete will continue to have access to the HSM until you reboot the client workstation

Set up IBM WebSphere MQ

IBM WebSphere MQ (Server or Client) must be installed on the target machines to carry on with the integration process. You can install IBM WebSphere MQ explorer for a Windows or Linux system. This graphical tool enables you to explore and configure all WebSphere MQ objects and resources and can remotely connect to queue managers on any supported platform. You also need to create the required user ID and group ID before you install WebSphere MQ. For a detailed installation procedure, refer to the IBM WebSphere MQ documentation.

https://www.ibm.com/docs/en/ibm-mq

Integrating Luna HSM with IBM WebSphere MQ

Luna HSM provides strong physical protection of secure assets, including keys, and should be considered a best practice when using IBM MQ. Following are the steps involved in this integration:

> Configuring SSL/TLS for IBM WebSphere MQ using Luna HSM

> Verifying SSL/TLS connectivity between MQ Server and MQ Client

Configure SSL/TLS for IBM WebSphere MQ using Luna HSM

To set up your SSL/TLS installation, you must start the queue manager, obtain and manage your digital certificates, and configure TLS to use Luna HSM.

> **NOTE:** On a test system, you can use self-signed certificates or certificates issued by a local certificate authority (CA). On a production system, use certificates issued by a trusted CA.

Following are the steps to configure IBM WebSphere MQ with Luna HSM:

> Create Queue Manager

> Create TLS certificates for MQ Server and MQ Client using Luna HSM

> Configure Queue Manager SSL/TLS to use Luna HSM Keys

> Configure MQ Client for SSL/TLS to use Luna HSM Keys

Create Queue Manager

Before you can use messages and queues, you must create and start at least one queue manager and its associated objects. To create and start the Queue Manager:

1. Log on to the IBM MQ Server as Administrator or a user added in mqm group.
2. Open The command prompt to run the following command for setting IBM MQ environment variables.

   ```
   $ . /opt/mqm/bin/setmqenv -s
   ```

3. If not created already, run the following command to create the queue manager.

   Windows:

   ```
   C:\Program Files\IBM\MQ\bin>crtmqm.exe QM1
   ```

   Linux:

   ```
   $ /opt/mqm/bin/crtmqm QM1
   ```

   Where `QM1` is queue manager name.

4. After creating the queue manager, start the queue manager using the following command:

   Windows:

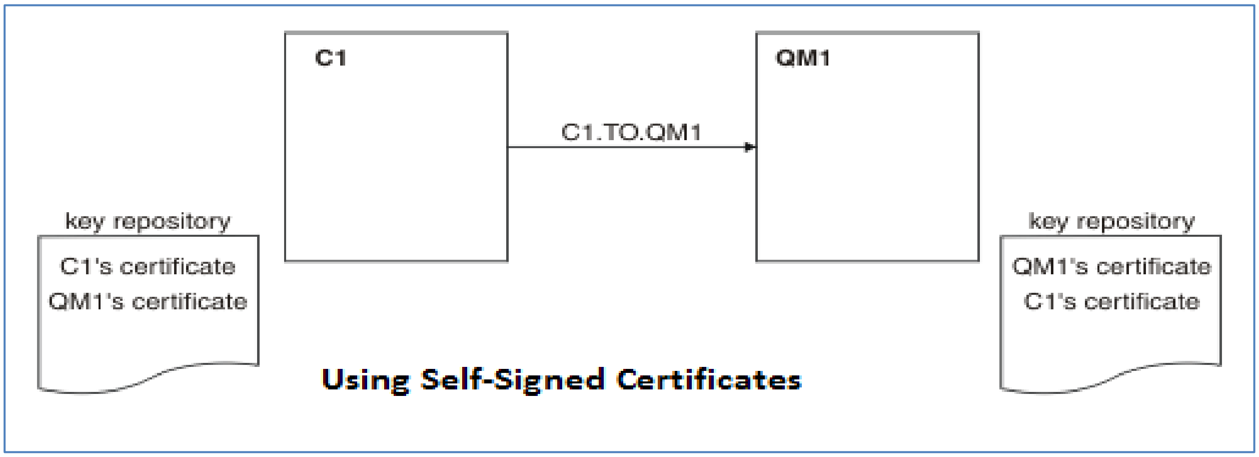
   ```
   C:\Program Files\IBM\MQ\bin>strmqm.exe QM1
   ```

   Linux:

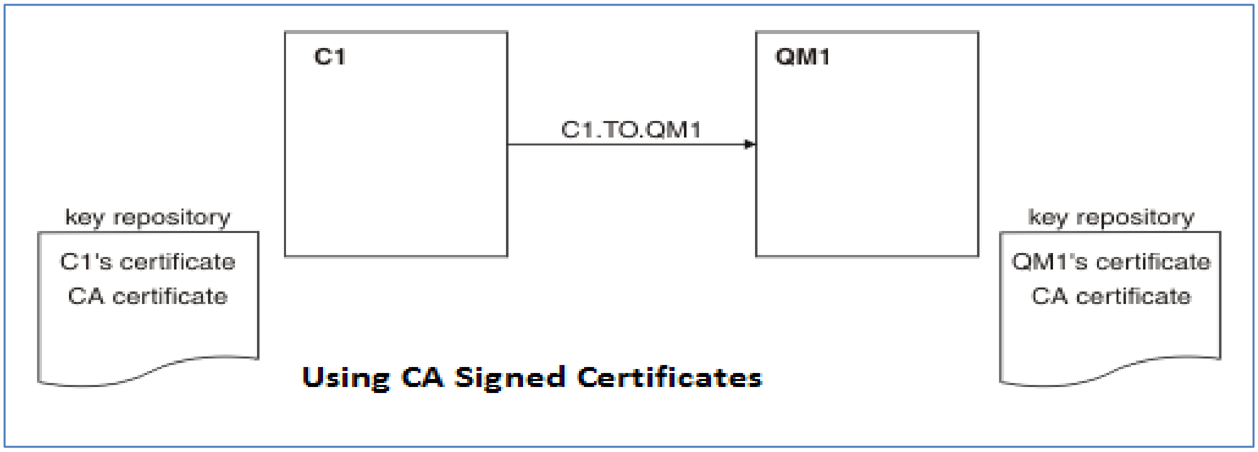
   ```
   $ /opt/mqm/bin/strmqm QM1
   ```

Create TLS certificates for MQ Server and MQ Client using Luna HSM

You can use a self-signed certificate or CA signed certificate as per your requirement. The steps provided here depict how to use a self-signed certificate. To use a CA signed certificate, you need to obtain the CA signed certificate and import both the CA certificate and the signed certificate into the key database.



In the above figure, the key repository for QM1 contains the certificate for QM1 and the public certificate from C1. The key repository for C1 contains the certificate for C1 and the public certificate from QM1.



In the above figure, the key repository for C1 contains certificate for C1 and the CA certificate. The key repository for QM1 contains the certificate for QM1 and the CA certificate.

To create the SSL/TLS certificate/key on Luna HSM:

> **NOTE:** These steps require to be executed on both MQ Server and MQ Client if you are securing both MQ Server and MQ Client SSL keys on Luna HSM.

1. Log on to the IBM MQ Server using the Administrator account or user added in the MQM group.
2. Create the `luna.cfg` file with the following contents and save the file at any location.

> **NOTE:** Path to Linux library = /usr/safenet/lunaclient/lib/libCryptoki2_64.so.

```
name = LUNA
library = C:\Program Files\SafeNet\LunaClient\cryptoki.dll
description = Luna config
slotListIndex = 0
attributes (*, CKO_PRIVATE_KEY, *) = {
CKA_SENSITIVE = true
CKA_SIGN=true
CKA_DECRYPT=true
}
attributes (*, CKO_PUBLIC_KEY, *) = {
CKA_VERIFY=true
CKA_ENCRYPT=true
}
attributes (*, CKO_SECRET_KEY, *) = {
CKA_SENSITIVE = true
CKA_ENCRYPT=true
CKA_DECRYPT=true
CKA_SIGN=true
CKA_VERIFY=true
}
disabledMechanisms = {
CKM_SHA1_RSA_PKCS
}
```

3. Open `<IBM MQ JRE installation path>\lib\security\java.security` file and edit IBMPKCS11 provider with the path to `luna.cfg` as follows:

> **NOTE:** Replace <Path to luna.cfg file> with actual path of luna.cfg file.

```
security.provider.1=com.ibm.fips.jsse.IBMJSSEFIPSProvider
security.provider.2=com.ibm.crypto.fips.provider.IBMJCEFIPS
security.provider.3=com.ibm.jsse2.IBMJSSEProvider2
security.provider.4=com.ibm.crypto.provider.IBMJCE
security.provider.5=com.ibm.security.jgss.IBMJGSSProvider
security.provider.6=com.ibm.security.cert.IBMCertPath
security.provider.7=com.ibm.crypto.pkcs11impl.provider.IBMPKCS11Impl
<Path to luna.cfg file>
security.provider.8=com.ibm.security.sasl.IBMSASL
security.provider.9=com.ibm.xml.crypto.IBMXMLCryptoProvider
security.provider.10=com.ibm.xml.enc.IBMXMLEncProvider
security.provider.11=com.ibm.security.jgss.mech.spnego.IBMSPNEGO
security.provider.12=sun.security.provider.Sun
security.provider.13=com.ibm.security.cmskeystore.CMSProvider
```

4. Run the `strmqikm.exe` from the `<IBM MQ installation directory>/bin` directory.

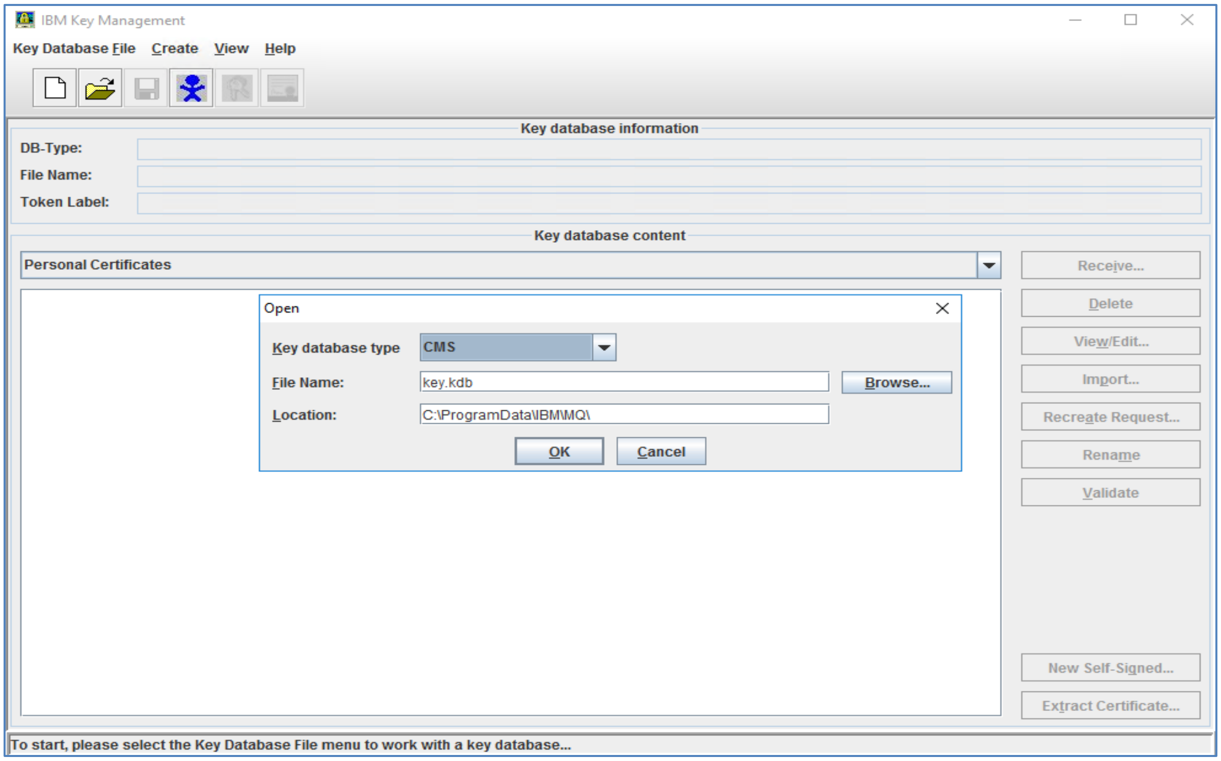
   Windows:

   ```
   C:\Program Files\IBM\MQ\bin>strmqikm.exe
   ```

   Linux:

   ```
   $ /opt/mqm/bin/strmqikm
   ```

5. When IBM Key Management windows pops up, click **Key Database File > Open**.

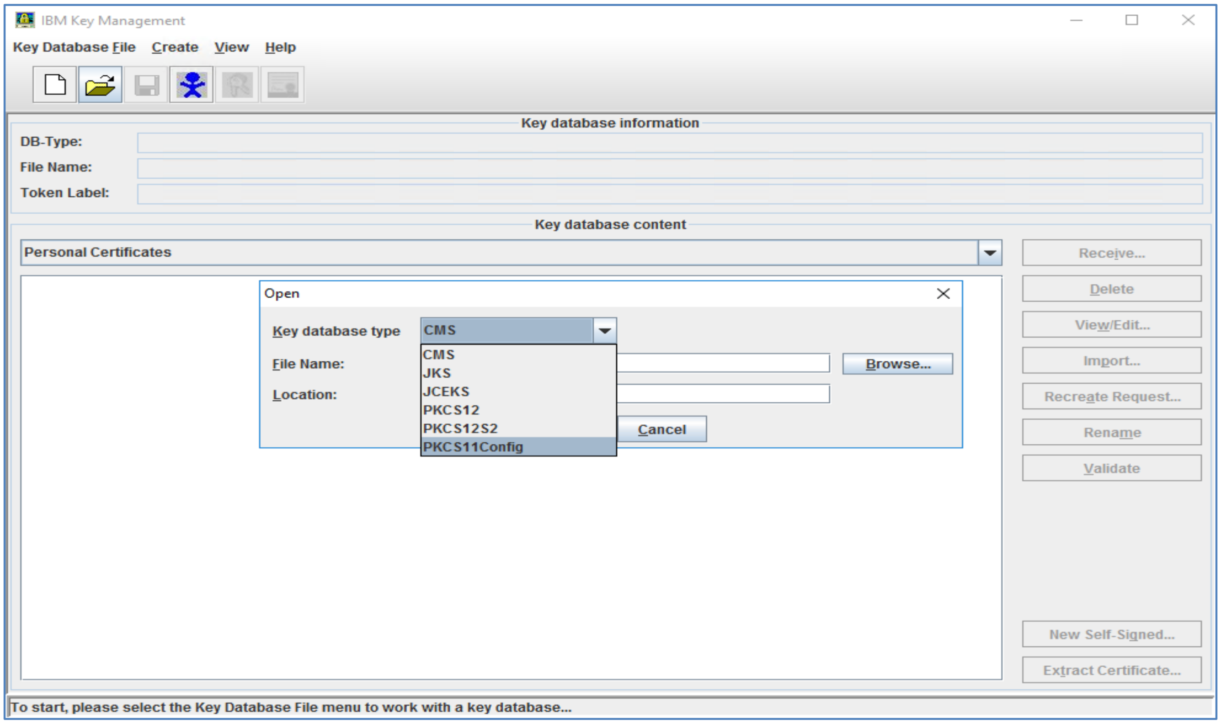


DB-Type:
File Name:
Token Label:

Key database type: CMS
File Name: key.kdb
Location: C:\ProgramData\IBM\MQ\

To start, please select the Key Database File menu to work with a key database...

6. Select **PKCS11Config** from Key database type drop-down and click **OK**.



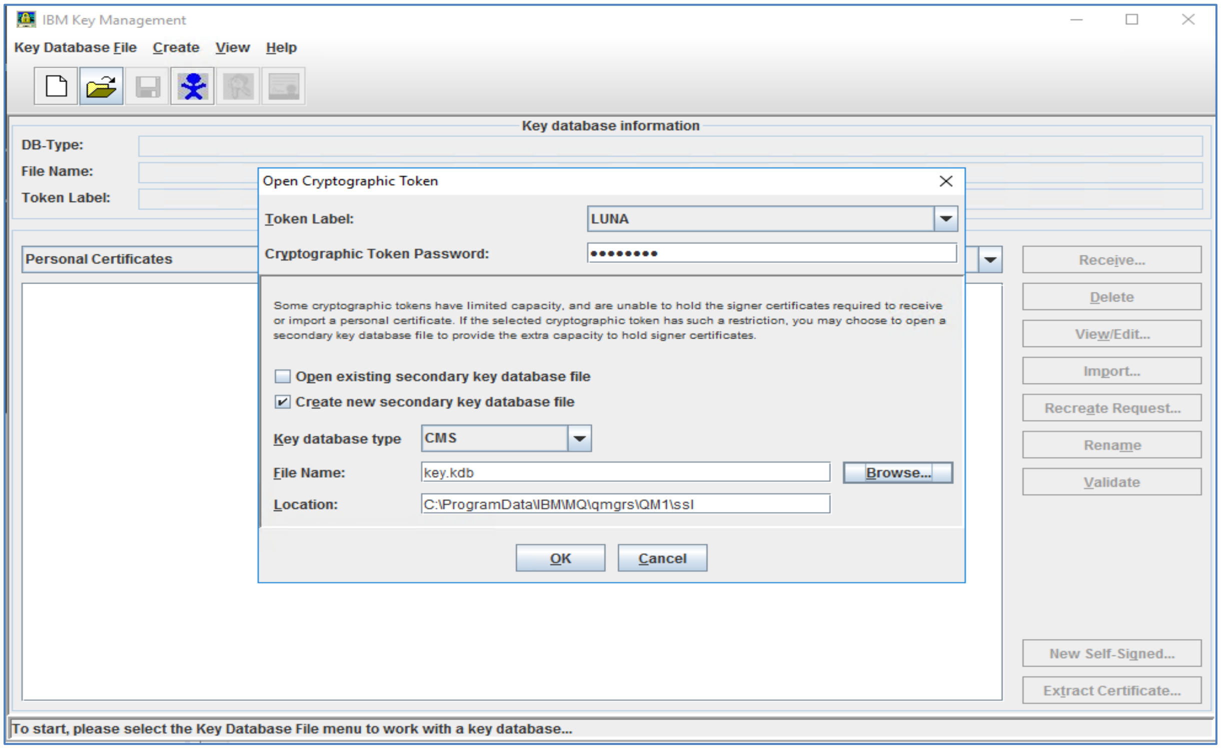
Key database type: CMS

File Name:

Location:

To start, please select the Key Database File menu to work with a key database...

7. Enter partition password in **Cryptographic Token Password**, select **Create new secondary key database** file, and then click **Browse…** to select the `key.kdb` file location. Click **OK.**



Token Label: LUNA

Cryptographic Token Password: ••••••••

Some cryptographic tokens have limited capacity, and are unable to hold the signer certificates required to receive or import a personal certificate. If the selected cryptographic token has such a restriction, you may choose to open a secondary key database file to provide the extra capacity to hold signer certificates.

☐ Open existing secondary key database file

☑ Create new secondary key database file

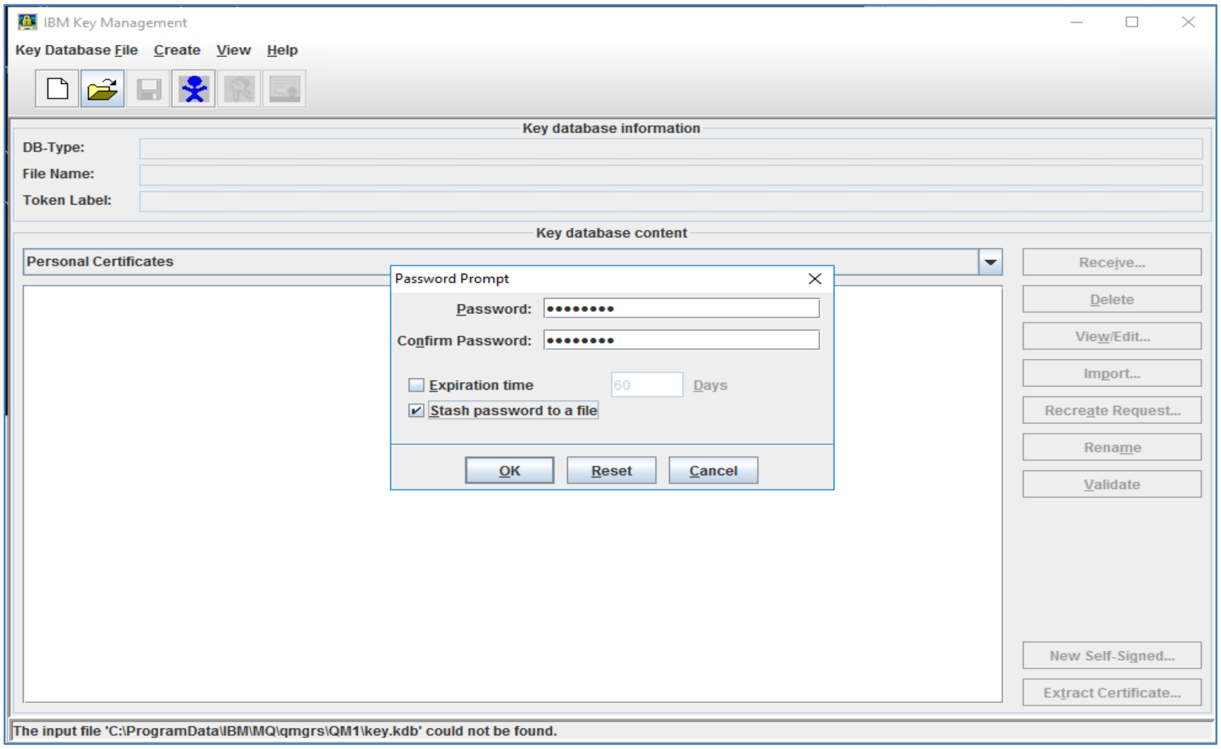
Key database type: CMS

File Name: key.kdb

Location: C:\ProgramData\IBM\MQ\qmgrs\QM1\ssl

To start, please select the Key Database File menu to work with a key database...

8. A password prompt window will pop up. Create a password for `key.kdb` file, provide the password and select the **Stash password to a file** checkbox, and then click **OK**.

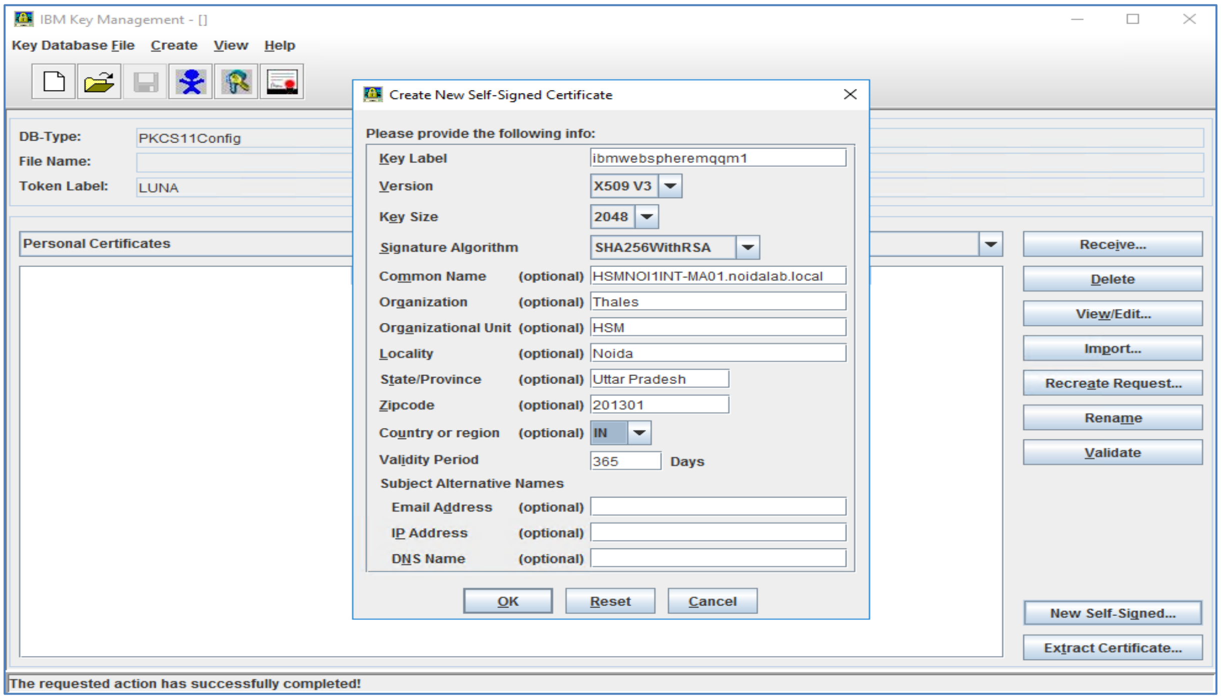


DB-Type:
File Name:
Token Label:
Personal Certificates

Password Prompt
Password: ••••••••
Confirm Password: ••••••••
☐ Expiration time 60 Days
☑ Stash password to a file

The input file 'C:\ProgramData\IBM\MQ\qmgrs\QM1\key.kdb' could not be found.

9. Click **Create > New Self-Signed Certificate**. Enter the **Key Label** in lowercase letters in the specific format “`ibmwebspheremq+queuename`”, where queuename is the name of the queue manager. Enter all other details for the certificate and then click **OK**.



DB-Type: PKCS11Config
File Name:
Token Label: LUNA
Personal Certificates

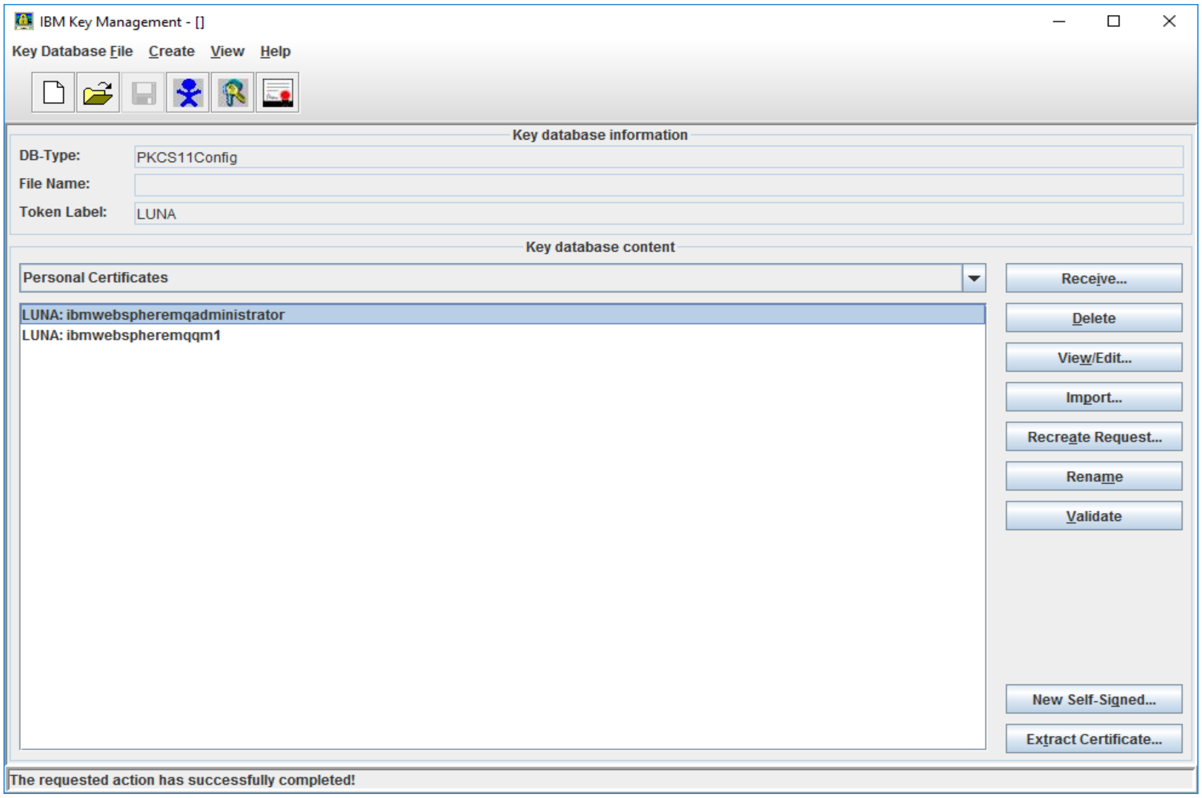
Create New Self-Signed Certificate

Please provide the following info:

Key Label		ibmwebspheremqqm1
Version		X509 V3
Key Size		2048
Signature Algorithm		SHA256WithRSA
Common Name	(optional)	HSMNOI1INT-MA01.noidalab.local
Organization	(optional)	Thales
Organizational Unit	(optional)	HSM
Locality	(optional)	Noida
State/Province	(optional)	Uttar Pradesh
Zipcode	(optional)	201301
Country or region	(optional)	IN
Validity Period		365 Days
Subject Alternative Names		
Email Address	(optional)	
IP Address	(optional)	
DNS Name	(optional)	

The requested action has successfully completed!

10. Log on to the IBM MQ Client using the administrator or user account added in the MQM group.
11. Repeat steps 2-9 for MQ Client to generate the IBM MQ Client certificate, while ensuring that IBM MQ Client Key Label also has the specific format "`ibmwebspheremq+<logged_in_user>`" with lowercase letters. Both MQ Server and Client Certificates are generated on the Luna HSM partition and are visible under Personal Certificates.



Key database information

DB-Type: PKCS11Config

File Name:

Token Label: LUNA

Key database content

Personal Certificates

LUNA: ibmwebspheremqadministrator

LUNA: ibmwebspheremqqm1

The requested action has successfully completed!

> **NOTE:** For demonstration purpose in this guide, the same partition and self-signed certificate is used for both MQ Server and Client, but it is recommended to use separate partition and CA signed certificates for MQ Server and Client.

12. Export the Server Certificate. To export the MQ Server certificate, select the certificate **under Personal Certificate** and click **Extract Certificate….** Then, save the certificate file and transfer the file to the MQ Client.
13. To import the Server Certificate on MQ Client, select **Signer Certificate** under CMS Key database content drop-down and click **Add….** Then, browse the server certificate you transferred, and click **OK** to add the Server certificate.
14. A window will pop up to provide a label for the certificate. Provide a certificate label and click **OK**.

Repeat steps 12-14 to export the Client certificate and add it to the Server's CMS Key database.

> **NOTE:** If you are using a CA signed certificate, you are required to add the CA public certificate of MQ Server in CMS Key database of MQ Client and vice-versa.

Configure Queue Manager SSL/TLS to use Luna HSM Keys

IBM MQ Server must be configured to use the Luna HSM for certificate and keys used to create the SSL/TLS connection with client. To configure the IBM MQ Server:

1. Log on to the IBM MQ Server using Administrator or user added in the mqm group.
2. Run the `MQExplorer.exe` from the `<IBM MQ Explorer installation>/bin` directory.

> **NOTE:** If you do not have MQ Explorer installed on MQ Server, proceed to **step 7** to add the Cryptographic Hardware settings using MQSC command-line.

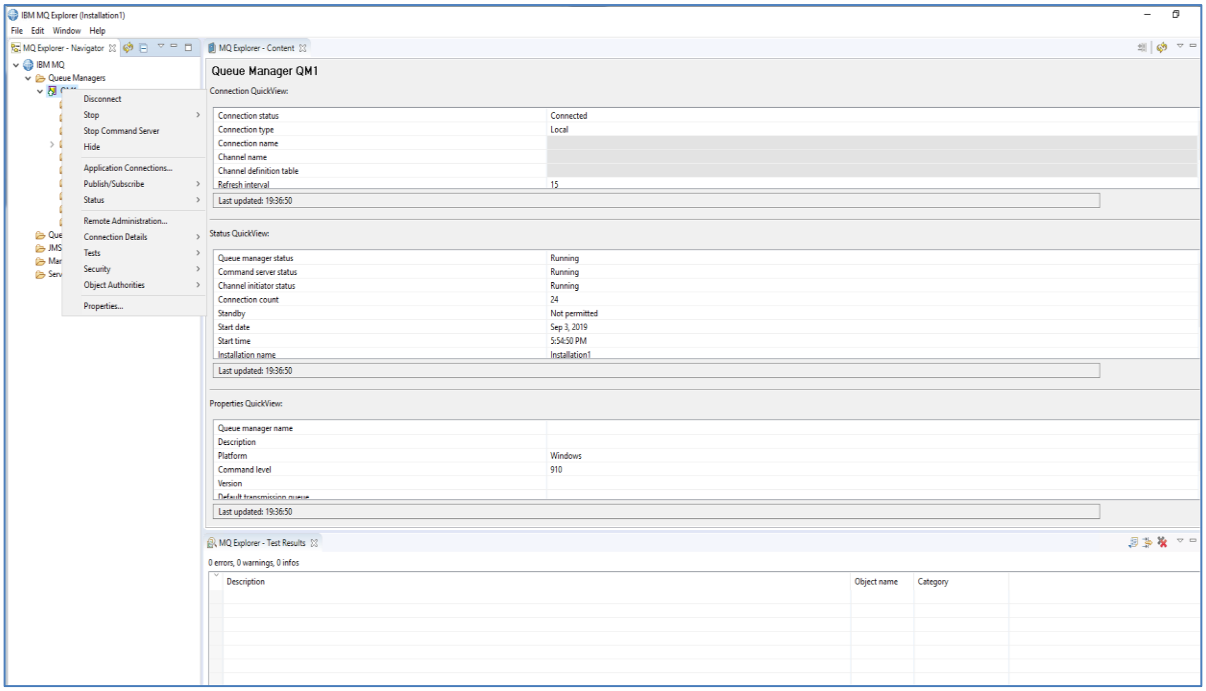
Windows:

```
C:\Program Files\IBM\MQ\bin>MQExplorer.exe
```

Linux:

```
$ /opt/mqm/bin/MQExplorer
```

3. In **MQ Explorer–Navigator** pane, expand **IBM MQ > Queue Managers**, right-click on queue manager you have created, and then click **Properties….**



Queue Manager QM1

Connection QuickView:

Connection status	Connected
Connection type	Local
Connection name	
Channel name	
Channel definition table	
Refresh interval	15

Last updated: 19:36:50

Status QuickView:

Queue manager status	Running
Command server status	Running
Channel initiator status	Running
Connection count	24
Standby	Not permitted
Start date	Sep 3, 2019
Start time	5:54:50 PM
Installation name	Installation1

Last updated: 19:36:50

Properties QuickView:

Queue manager name	
Description	
Platform	Windows
Command level	910
Version	
Default transmission queue	

Last updated: 19:36:50

0 errors, 0 warnings, 0 infos

Description	Object name	Category

4. In **Queue Manager-Properties** window, click **SSL** from the menu on the right side and then **Configure…** under the Cryptographic hardware section. Select **other (PKCS11)** and enter the **Driver path**:

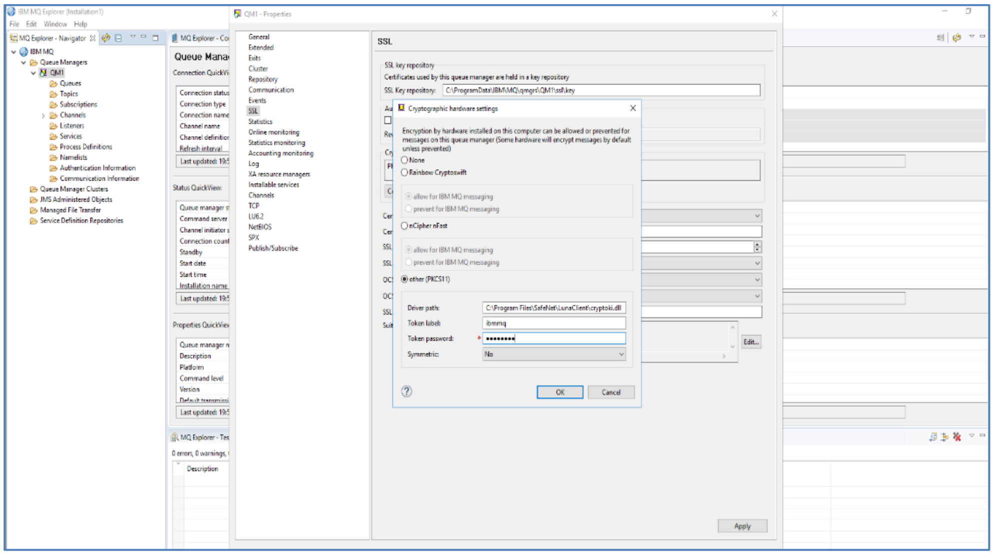
   <u>Windows:</u>

   ```
   C:\Program Files\SafeNet\LunaClient\cryptoki.dll
   ```

   <u>Linux:</u>

   ```
   /usr/safenet/lunaclient/lib/libCryptoki2_64.so
   ```

   followed by **Token label** and **Token password**. Click **OK** and then click **Apply** to save the settings. When details are saved, close the SSL properties window.



Cryptographic hardware settings

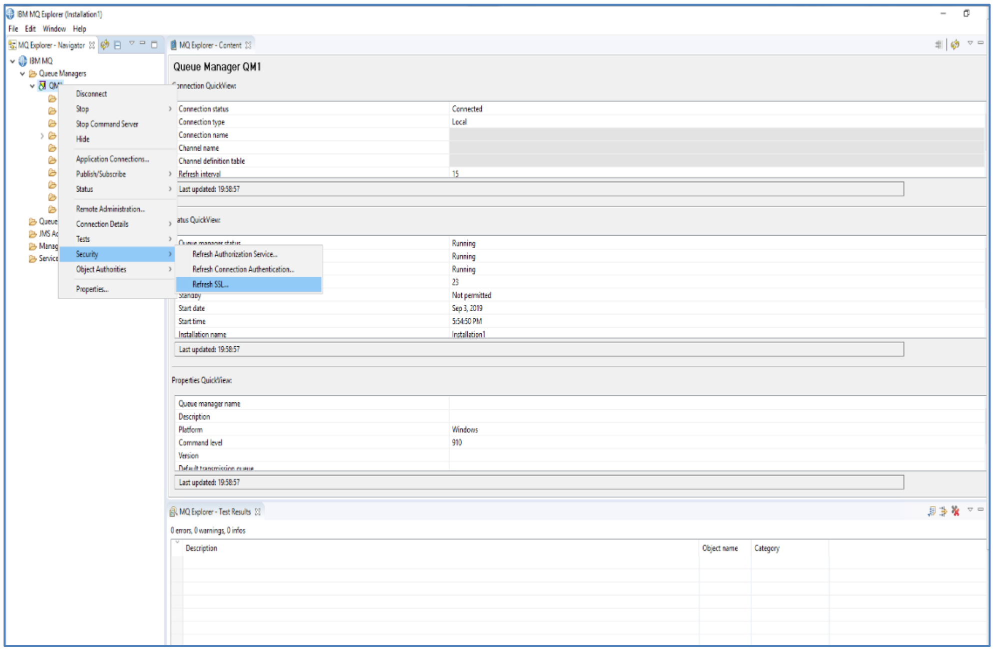
Encryption by hardware installed on this computer can be allowed or prevented for messages on this queue manager (Some hardware will encrypt messages by default unless prevented)

○ None
○ Rainbow Cryptoswift
○ nCipher nFast
● other (PKCS11)

Driver path: C:\Program Files\SafeNet\LunaClient\cryptoki.dll
Token label: ibmmq
Token password: ••••••••
Symmetric: No

SSL Key repository: C:\ProgramData\IBM\MQ\qmgrs\QM1\ssl\key

5. Right Click on Queue Manager and click **Security > Refresh SSL….**



Connection QuickView:	
Connection status	Connected
Connection type	Local
Connection name	
Channel name	
Channel definition table	
Refresh interval	15

Last updated: 19:58:57

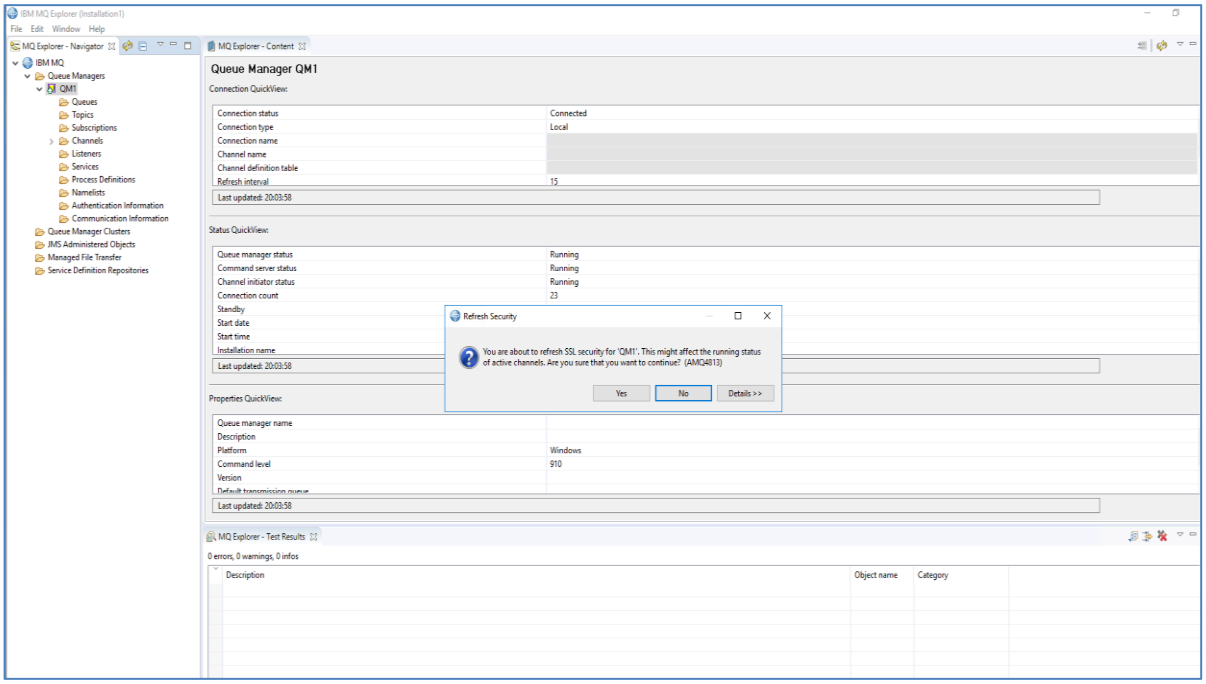
Status QuickView:	
Queue manager status	Running
	Running
	Running
	23
Standby	Not permitted
Start date	Sep 3, 2019
Start time	5:54:50 PM
Installation name	Installation1

Last updated: 19:58:57

Properties QuickView:	
Queue manager name	
Description	
Platform	Windows
Command level	910
Version	
Default transmission queue	

Last updated: 19:58:57

6. A message will pop up to ensure that you are refreshing the SSL security. Click **Yes** to refresh the SSL settings.



Queue Manager QM1

Connection QuickView:

Connection status	Connected
Connection type	Local
Connection name	
Channel name	
Channel definition table	
Refresh interval	15

Last updated: 20:03:58

Status QuickView:

Queue manager status	Running
Command server status	Running
Channel initiator status	Running
Connection count	23
Standby	
Start date	
Start time	
Installation name	

Last updated: 20:03:58

Properties QuickView:

Queue manager name	
Description	
Platform	Windows
Command level	910
Version	
Default transmission queue	

Last updated: 20:03:58

0 errors, 0 warnings, 0 infos

Description	Object name	Category

Refresh Security

You are about to refresh SSL security for 'QM1'. This might affect the running status of active channels. Are you sure that you want to continue? (AMQ4813)

7. Open the MQSC console to run the MQSC commands for the Queue Manager.

> **NOTE:** Proceed to **step 12**, if you have already enabled cryptographic hardware settings using MQ Explorer.

Windows:

```
C:\Program Files\IBM\MQ\bin>runmqsc.exe QM1
```

Linux:

```
$ /opt/mqm/bin/runmqsc QM1
```

> **Note:** Replace `QM1` with your queue manager's name.

8. Modify QMGR for SSLCRYP (Cryptographic Hardware) settings using the following command format:

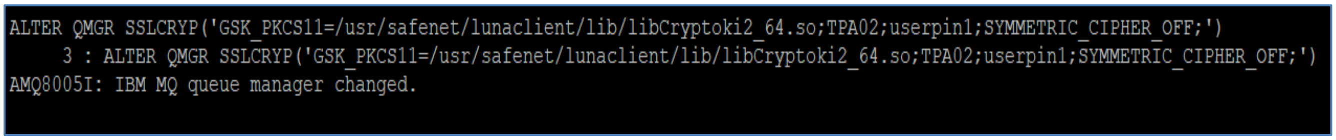
```
ALTER QMGR SSLCRYP (string)
```

The `string` should adhere to this specific structure:

```
“GSK_PKCS11=<the PKCS #11 driver path and file name>;<the PKCS #11 token
label>;<the PKCS #11 token password>;symmetric cipher setting”
```

Example:

```
ALTER QMGR
SSLCRYP('GSK_PKCS11=/usr/safenet/lunaclient/lib/libCryptoki2_64.so;TPA02;us
erpin1;SYMMETRIC_CIPHER_OFF;')
```



```
ALTER QMGR SSLCRYP('GSK_PKCS11=/usr/safenet/lunaclient/lib/libCryptoki2_64.so;TPA02;userpin1;SYMMETRIC_CIPHER_OFF;')
     3 : ALTER QMGR SSLCRYP('GSK_PKCS11=/usr/safenet/lunaclient/lib/libCryptoki2_64.so;TPA02;userpin1;SYMMETRIC_CIPHER_OFF;')
AMQ8005I: IBM MQ queue manager changed.
```

9. Modify QMGR for SSLKEYR (Key Repository) settings using the following command format:

   `ALTER QMGR SSLKEYR (string)`

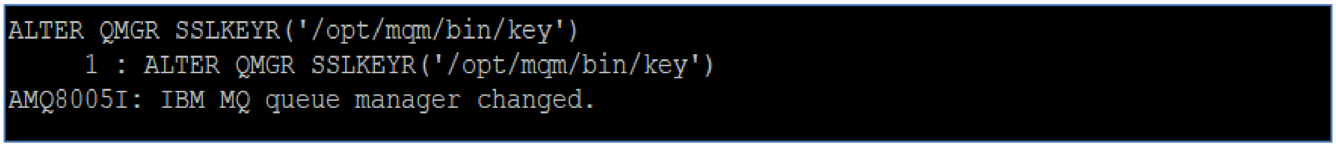
   The `string` should be structured as follows:

   `pathname/keyfile`

   The `keyfile` should specify the GSKit CMS key database file without the “.kdb” suffix.

   Example:

   `ALTER QMGR SSLKEYR('/opt/mqm/bin/key')`

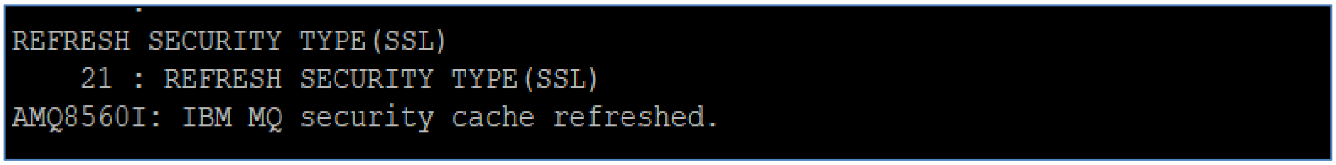


```
ALTER QMGR SSLKEYR('/opt/mqm/bin/key')
     1 : ALTER QMGR SSLKEYR('/opt/mqm/bin/key')
AMQ8005I: IBM MQ queue manager changed.
```

NOTE: Ensure to replace placeholders with your actual values.

10. Once the changes are made, refresh the queue manager SSL settings.

    `REFRESH SECURITY TYPE(SSL)`



```
REFRESH SECURITY TYPE(SSL)
    21 : REFRESH SECURITY TYPE(SSL)
AMQ8560I: IBM MQ security cache refreshed.
```

11. Type `end` and press ENTER to close the MQSC console.

12. Run MQSC commands for queue manager to verify the SSL settings.

    Windows:

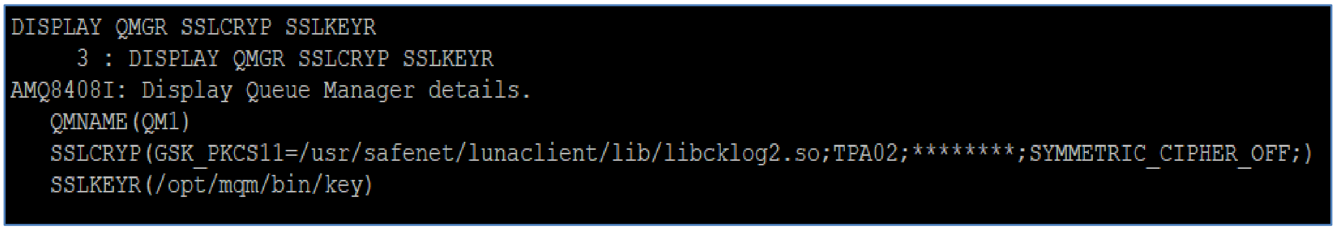
    `C:\Program Files\IBM\MQ\bin>runmqsc.exe QM1`

    Linux:

    `$ /opt/mqm/bin/runmqsc QM1`

13. Run `DISPLAY QMGR` and verify the SSLCRYP and SSLKEYR value.

    `DISPLAY QMGR SSLCRYP SSLKEYR`



```
DISPLAY QMGR SSLCRYP SSLKEYR
     3 : DISPLAY QMGR SSLCRYP SSLKEYR
AMQ8408I: Display Queue Manager details.
   QMNAME(QM1)
   SSLCRYP(GSK_PKCS11=/usr/safenet/lunaclient/lib/libcklog2.so;TPA02;********;SYMMETRIC_CIPHER_OFF;)
   SSLKEYR(/opt/mqm/bin/key)
```

14. On QM1 queue manager, create a listener for the connection channel.

```
DEFINE LISTENER(QM1.TCP) TRPTYPE(TCP) PORT(1414) CONTROL(QMGR)
```

Where `LISTENER` is the user-defined name, such as `QM1.TCP`.

15. Start the listener.

```
START LISTENER(QM1.TCP)
```

16. Check the status of listener.

```
DISPLAY LSSTATUS(QM1.TCP)
```

17. Create server connection channel.

Windows:

```
DEFINE CHANNEL(C1.TO.QM1) CHLTYPE(SVRCONN) TRPTYPE(TCP)
MCAUSER('MUSR_MQADMIN') SSLCAUTH(REQUIRED)
SSLCIPH(TLS_RSA_WITH_AES_128_CBC_SHA256) DESCR('Server channel using SSL
from C1 to QM1')
```

Linux:

```
DEFINE CHANNEL(C1.TO.QM1) CHLTYPE(SVRCONN) TRPTYPE(TCP) MCAUSER('mqm')
SSLCAUTH(REQUIRED) SSLCIPH(TLS_RSA_WITH_AES_128_CBC_SHA256) DESCR('Server
channel using SSL from C1 to QM1')
```

Where:

- `CHANNEL` is the user-defined channel name.
- `MCAUSER` is the user added in the MQM group.

18. Start the channel.

```
START CHANNEL(C1.TO.QM1)
```

19. Create a client connection channel.

```
DEFINE CHANNEL(C1.TO.QM1) CHLTYPE(CLNTCONN) TRPTYPE(TCP)
CONNAME('HSMNOI1INT-MA01.noidalab.local') QMNAME(QM1)
SSLCIPH(TLS_RSA_WITH_AES_128_CBC_SHA256) DESCR('Client channel using SSL
from C1 to QM1')
```

Here, `CONNAME` is the hostname or IP of the MQ Server.

> **NOTE:** Ensure that channel name and SSL cipher for client connection channel must match the server connection channel.

20. Create channel authentication record to authorize the client when it connects to MQ Server.

Windows:

```
SET CHLAUTH(C1.TO.QM1) TYPE(ADDRESSMAP) ADDRESS('HSMNOI1INT-
MA02.noidalab.local') MCAUSER('MUSR_MQADMIN')
```

Linux:

```
SET CHLAUTH(C1.TO.QM1) TYPE(ADDRESSMAP) ADDRESS('HSMNOI1INT-
MA02.noidalab.local') MCAUSER('mqm')
```

Where:

- `ADDRESS` is the hostname or IP of the MQ Client.

- `MCAUSER` is the user authorized to use channel.

> **NOTE:** For channel authentication type other than `ADDRESSMAP`, refer the IBM MQ Documentation. In case all channels are blocked for admin user by default rules, create an override rule for admin user to access the channel.
>
> ```
> SET CHLAUTH(C1.TO.QM1) TYPE(BLOCKUSER) DESCR('Rule to override
> *MQADMIN blockuser on this channel') USERLIST('nobody')
> ACTION(replace)
> ```

21. Create a local queue on QM1.

    ```
    DEFINE QLOCAL(QM1.LQ)
    ```

    Here, `QM1.LQ` is the user defined local queue.

22. To close MQSC, type `END` and then press Enter.

23. Restart the queue manager to apply all the changes. To restart, run the following commands:

    Windows:

    ```
    C:\Program Files\IBM\MQ\bin>endmqm.exe QM1
    C:\Program Files\IBM\MQ\bin>strmqm.exe QM1
    ```

    Linux:

    ```
    $ /opt/mqm/bin/endmqm QM1
    $ /opt/mqm/bin/strmqm QM1
    ```

Configure MQ Client for SSL/TLS to use Luna HSM Keys

MQ Client SSL keys and certificate are already generated on the Luna HSM. Now you need to configure the MQ Client to use these keys and certificate while establishing the SSL connection with MQ Server. To configure the client, perform the following steps:

1. Log on to the IBM MQ Client using the Administrator account or user added in mqm group.

2. To configure Hardware Cryptographic Token, add the SSL stanza in the `mqclient.ini` file available at the `C:\ProgramData\IBM\MQ` for Windows and `/var/mqm` directory in Linux.

   Windows:

   ```
   SSL:
       SSLCryptoHardware=GSK_PKCS11=C:\Program
       Files\SafeNet\LunaClient\cryptoki.dll;ibmmq;userpin1;SYMMETRIC_CIPHER_OFF
       SSLKeyRepository=C:\ProgramData\IBM\MQ\key
   ```

   Linux:

   ```
   SSL:
       SSLCryptoHardware=GSK_PKCS11=/usr/safenet/lunaclient/lib/libCryptoki2_64
       .so\;ibmmq\;userpin1\;SYMMETRIC_CIPHER_OFF
       SSLKeyRepository=/var/mqm/key
   ```

   Where:

- `SSLCryptoHardware` is a string in the format: `GSK_PKCS11 = driver path and filename;token label;token password;symmetric cipher setting` and
- `SSLKeyRepository` is the location of key repository that holds the user's digital certificate in stem format. That is, it includes the full path and the file name without an extension.

3. Copy the Client Channel Definition Table (`AMQCLCHL.TAB`) from MQ Server to MQ Client.

   The Client Channel Definition Table is located at the following location on the MQ server:

   Windows:

   ```
   C:\ProgramData\IBM\MQ\qmgrs\QM1\@ipcc\
   ```

   Linux:

   ```
   /var/mqm/qmgrs/QM1/@ipcc/
   ```

   Copy the `AMQCLCHL.TAB` file to MQ Client at `"C:\ProgramData\IBM\MQ"` or `"/var/mqm"` for Windows and Linux platform respectively.

Configuring SSL/TLS for IBM WebSphere MQ using Luna HSM

The TLS (Transport Layer Security) protocol enables queue managers to communicate securely with other queue managers or clients. When a queue manager connects with another queue manager or client, the two carry out a standard TLS exchange of certificates and validation checks. If the validation is successful, the connection is established. To achieve this, the queue managers and clients, as well as the channels that they use, must be configured with appropriate certificate settings.

When messages are sent from one queue manager to another queue manager or client along a channel, the data is generally encrypted using a session key that has been established during the certificate exchange. To achieve this, you must configure the channels that you use with appropriate cipher specs.

You have already generated the certificates on Luna HSM and have configured both queue manager and clients to use the TLS connection through the channel created and authenticated using Channel Authentication Record. To verify SSL/TLS connectivity between MQ server and MQ client:

1. Log on to the IBM MQ Client through the Administrator account or a user added to mqm group.
2. Open the command prompt and set the following environment variables:

   Windows:

   ```
   set MQCHLLIB=C:\ProgramData\IBM\MQ
   set MQCHLTAB=AMQCLCHL.TAB
   set MQSAMP_USER_ID=MUSR_MQADMIN
   ```

   Linux:

   ```
   export MQCHLLIB=/var/mqm
   export MQCHLTAB=AMQCLCHL.TAB
   export MQSAMP_USER_ID=mqm
   ```

   Where:

   - `MQCHLLIB` is the location of the Client Channel Definition Table.

- `MQCHLTAB` is the Client Channel Definition Table copied from the MQ Server.
- `MQSAMP_USER_ID` is the user authorized in MQ Server channel authentication record.

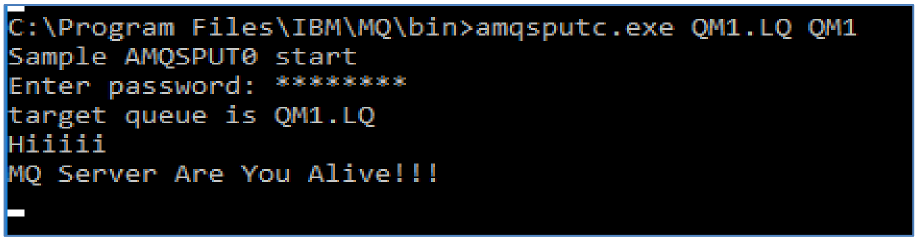
3. Run the following command to create the SSL connection from client to server and put messages in the MQ Server queue.

   Windows:

   ```
   C:\Program Files\IBM\MQ\bin>amqsputc.exe QM1.LQ QM1
   ```

   Linux:

   ```
   /opt/mqm/samp/bin/amqsputc QM1.LQ QM1
   ```

   ```
   C:\Program Files\IBM\MQ\bin>amqsputc.exe QM1.LQ QM1
   Sample AMQSPUT0 start
   Enter password: ********
   target queue is QM1.LQ
   Hiiiii
   MQ Server Are You Alive!!!
   ```

   The messages you type here will be delivered to MQ Server securely using the established SSL/TLS connection. MQ Server can access the messages received from MQ Client.

4. Log on to IBM MQ Server using the Administrator account or a user added to mqm group.

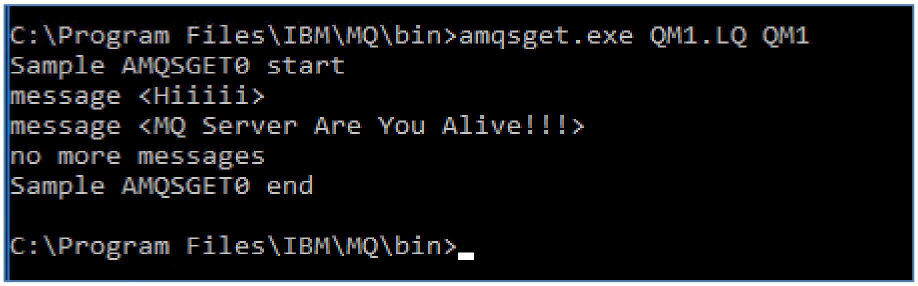
5. Open the command prompt and run the following command to get the message sent by client:

   Windows:

   ```
   C:\Program Files\IBM\MQ\bin>amqsget.exe QM1.LQ QM1
   ```

   Linux:

   ```
   /opt/mqm/samp/bin/amqsget QM1.LQ QM1
   ```

   ```
   C:\Program Files\IBM\MQ\bin>amqsget.exe QM1.LQ QM1
   Sample AMQSGET0 start
   message <Hiiiii>
   message <MQ Server Are You Alive!!!>
   no more messages
   Sample AMQSGET0 end

   C:\Program Files\IBM\MQ\bin>
   ```

   This completes the integration of IBM WebSphere MQ with Luna HSM by securing the SSL/TLS authentication certificates/keys on Luna HSM.

Contacting Customer Support

If you encounter a problem while installing, registering, or operating this product, refer to the documentation. If you cannot resolve the issue, contact your supplier or Thales Customer Support. Thales Customer Support operates 24 hours a day, 7 days a week. Your level of access to this service is governed by the support plan arrangements made between Thales and your organization. Please consult this support plan for further information about your entitlements, including the hours when telephone support is available to you.

Customer Support Portal

The Customer Support Portal, at https://supportportal.thalesgroup.com, is a database where you can find solutions for most common problems. The Customer Support Portal is a comprehensive, fully searchable repository of support resources, including software and firmware downloads, release notes listing known problems and workarounds, a knowledge base, FAQs, product documentation, technical notes, and more. You can also use the portal to create and manage support cases.

> **NOTE:** You require an account to access the Customer Support Portal. To create a new account, go to the portal and click on the **REGISTER** link.

Telephone Support

If you have an urgent problem, or cannot access the Customer Support Portal, you can contact Thales Customer Support by telephone at +1 410-931-7520. Additional local telephone support numbers are listed on the support portal.