IBM DB2

Integration Guide

gemalto
security to be free

Document Part Number: 007-013741-001, Rev. B

Release Date: November 2018

Contents

Preface

Scope

This document outlines the steps to integrate IBM DB2 with SafeNet Luna HSM. SafeNet Luna HSM is used to secure the Master Encryption Key for IBM DB2.

Document Conventions

This section provides information on the conventions used in this template.

Notes

Notes are used to alert you to important or helpful information. These elements use the following format:

NOTE: Take note. Contains important or helpful information.

Cautions

Cautions are used to alert you to important information that may help prevent unexpected results or data loss. These elements use the following format:

CAUTION: Exercise caution. Caution alerts contain important information that may help prevent unexpected results or data loss.

Warnings

Warnings are used to alert you to the potential for catastrophic data loss or personal injury. These elements use the following format:

WARNING: Be extremely careful and obey all safety and security measures. In this situation you might do something that could result in catastrophic data loss or personal injury.

Command Syntax and Typeface Conventions

Convention	Description
bold	The bold attribute is used to indicate the following: • Command-line commands and options (Type **dir /p**.)

Convention	Description
	• Button names (Click **Save As**.) • Check box and radio button names (Select the **Print Duplex** check box.) • Window titles (On the **Protect Document** window, click **Yes**.) • Field names (**User Name:** Enter the name of the user.) • Menu names (On the **File** menu, click **Save**.) (Click **Menu** > **Go To** > **Folders**.) • User input (In the **Date** box, type **April 1**.)
italic	The italic attribute is used for emphasis or to indicate a related document. (See the *Installation Guide* for more information.)
`Consolas`	Denotes syntax, prompts, and code examples.

Support Contacts

Contact Method	Contact Information	
Address	Gemalto 4690 Millennium Drive Belcamp, Maryland 21017, USA	
Phone	US	1-800-545-6608
	International	1-410-931-7520
Technical Support Customer Portal	https://supportportal.gemalto.com Existing customers with a Technical Support Customer Portal account can log in to manage incidents, get the latest software upgrades, and access the Gemalto Knowledge Base.	

1
Introduction

Overview

This document is intended to guide security administrators through the steps for the IBM DB2 Integration with SafeNet Luna HSM, and also covers the necessary information to install, configure and integrate IBM DB2 with SafeNet Luna HSM.

IBM DB2 is the database of choice for enterprise-wide solutions Optimized to deliver industry-leading performance while lowering costs, IBM DB2 offers extreme performance, flexibility, scalability and reliability for any size organization.

One can encrypt the databases and backup images using DB2 native encryption. Native encryption provides transparent and secure key management and requires no changes to your hardware, software, applications, or schemas.

Master keys are stored in a file or database called a keystore. A user can use one of the following keystores:

- A local keystore that is located on the same system as the DB2 server.
- A centralized keystore that is located on a system other than the DB2 server.
- A PKCS #11 keystore that is located on a system other than the DB2 server.

The primary benefit of a PKCS #11 keystore is the protection it provides to encryption keys. This protection is accomplished by imposing a restriction that keys never leave the secure environment of the keystore. Data on disk is encrypted with a data encryption key (DEK) that is stored with the database. The DEK, in turn, is encrypted by a master key (MK), which is stored externally to the database. The DEK is sent to the PKCS #11 keystore, where it is decrypted by the MK. The only exception to this principle of keys not leaving the keystore is when migrating keys from a local keystore file to a PKCS #11 keystore. In such cases, these keys are marked as external. However, an immediate key rotation following migration will start to make use of internally defined keys.

- Using a centralized keystore or a PKCS #11 keystore is useful when you have multiple databases and you do not want to maintain individual keystores.
- A password is required to access the keystore. You can optionally store (or stash) the keystore password, in obfuscated form, in a stash file. Stashing the password makes it possible for the database manager to access the keystore (when the database manager starts, for example) without someone having to manually enter the keystore password.
- For use with native encryption, a local keystore must be compliant with the public-key cryptography standard 12 (PKCS#12).
- A DB2 instance can be configured to use only one keystore for native encryption at a time.

The following are the benefits of using SafeNet HSMs to secure the IBM DB2 key:

- Secure generation, storage and protection of the Identity signing private key on FIPS 140-2 level 3 validated hardware.
- Full life cycle management of the keys.
- HSM audit trail.
- Take advantage of cloud services with confidence.
- Significant performance improvements by off-loading cryptographic operations from application servers.

Third Party Application Details

This integration guide uses the following third party application details.

- IBM DB2

Supported Platforms

Below is the list of the platforms tested with the following HSMs:

SafeNet Luna HSM: SafeNet 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. SafeNet Luna HSMs physically and logically secure cryptographic keys and accelerate cryptographic processing.

The SafeNet Luna HSM on premise offerings include the SafeNet Luna Network HSM, SafeNet PCIe HSM, and SafeNet Luna USB HSMs. SafeNet Luna HSMs are also available for access as an offering from cloud service providers such as IBM cloud HSM and AWS cloud HSM classic.

This integration is supported with SafeNet Luna HSM on the following operating systems:

- RHEL
- CentOS
- AIX

SafeNet DPOD: SafeNet Data Protection on Demand (DPoD) is a cloud-based platform that provides on-demand HSM and Key Management services through a simple graphical user interface. With DPOD, security is simple, cost effective, and easy to manage because there is no hardware to buy, deploy, and maintain. As an Application Owner, you click and deploy services, generate usage reports and maintain the required services.

This integration is supported with SafeNet DPOD on the following operating systems:

- RHEL
- CentOS
- AIX

Prerequisites

Before you proceed with the integration, ensure that you complete Configuring the SafeNet Luna Network HSM or Provision your HSM on Demand service as per the integration requirement.

Configuring SafeNet Luna HSM

Before you get started ensure the following:

1. Ensure the HSM is setup, initialized, provisioned and ready for deployment. Refer to the HSM product documentation for help.
2. Create a partition on the HSM that will be later used by IBM DB2.
3. If using a SafeNet Luna Network HSM, register a client for the system and assign the client to the partition to create an NTLS connection. Initialize Crypto Officer and Crypto User roles for the registered partition. Ensure that the partition is successfully registered and configured. The command to see the registered partition is:

```
<Path to lunacm utility>lunacm
LunaCM v7.1.0-379. Copyright (c) 2006-2017 SafeNet.


Available HSMs:
Slot Id ->              0
Label ->                IBMDB2
Serial Number ->        1238712343066
Model ->                LunaSA 7.1.0
Firmware Version ->     7.1.0
Configuration ->        Luna User Partition With SO (PED) Key Export With Cloning Mode
Slot Description ->     Net Token Slot
```

NOTE: Follow the SafeNet Network Luna HSM documentation for detailed steps for creating NTLS connection, initializing the partitions and initializing the necessary user roles.

Provision your HSM on Demand service

This service provides your client machine with access to an HSM Application Partition for storing cryptographic objects used by your applications. Application partitions can be assigned to a single client, or multiple clients can be assigned to, and share, a single application partition.

You need to provision your application partition, starting by initializing the following roles:

- **Security Officer (SO)** - responsible for setting the partition policies and for creating the Crypto Officer.
- **Crypto Officer (CO)** - responsible for creating, modifying and deleting crypto objects within the partition. The CO can use the crypto objects and create an optional, limited-capability role called Crypto User that can use the crypto objects but cannot modify them.
- **Crypto User (CU)** - optional role that can use crypto objects while performing cryptographic operations.

NOTE: Refer the “SafeNet Data Protection on Demand Application Owner Quick Start Guide” for configuring the HSM on Demand service and create a service client.

The HSM service client package is a zip file that contains system information needed to connect your client machine to an existing HSM on Demand service.

Constraints on HSM on Demand Services

Please consider the following if integrating an HSMoD service with Microsoft Active Directory Certificate Services.

HSM on Demand Service in FIPS mode

HSMoD services operate in a FIPS and non-FIPS mode. If your organization requires non-FIPS algorithms for your operations, ensure you enable the **Allow non-FIPS approved algorithms** check box when configuring your HSM on Demand service. The FIPS mode is enabled by default.

Refer to the “Mechanism List” in the SDK Reference Guide for more information about available FIPS and non-FIPS algorithms.

Verify HSM on Demand <slot> value

LunaCM commands work on the current slot. If there is only one slot, then it is always the current slot. If you are completing an integration using HSMoD services, you need to verify which slot on the HSMoD service you send the commands to. If there is more than one slot, then use the **slot set** command to direct a command to a specified slot. You can use slot list to determine which slot numbers are in use by which HSMoD service.

IBM DB2 Prerequisites for Installation

Before installing data server products on Linux operating systems, ensure that the system chosen meets the necessary operating system, hardware, software, and communications requirements.

For the installation requirements for data server products, see Appendix.

IBM DB2 Setup

IBM DB2 must be installed on the target machine to carry on with the integration process. For a detailed installation procedure of IBM DB2 refer to the IBM DB2 Documentation, reference of the same is provided in Appendix.

2

Integrating SafeNet Luna HSM with IBM DB2

Generating a Master Encryption Key for HSM-Based Encryption

To start using HSM-based encryption, you need to have a master encryption key that will be stored inside the HSM. The master encryption key is used to encrypt or decrypt column/tablespace encryption keys inside the HSM. HSM can be used in the following ways to protect the Master Encryption Key:

- An existing Master Encryption Key can be migrated onto the HSM.
- A Master Encryption Key can be directly generated onto the HSM.

Migrating Master Encryption Key onto the HSM

In order to migrate a Master Encryption Key for HSM-Based Encryption, perform the following instructions:

NOTE: It is assumed that no software-based wallet is yet created.

Verifying GSKit Installation and Configuration

To use DB2 native encryption, you must verify that GSKit is installed and configured.

On Linux and UNIX operating systems, the DB2 installer installs GSKit locally. For each instance, the GSKit libraries will be located in `sqllib/lib32/gskit` or `sqllib/lib64/gskit.`

Set the environment variable using below command:

```
./home/<db2_instance>/sqllib/db2profile
```

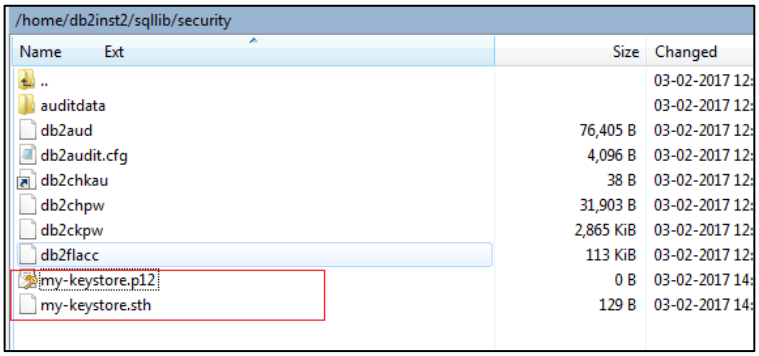
Creating Local keystore

Log in as the DB2 instance owner, traverse to the gskit directory: `/home/<Instance_owner>/sqllib/gskit/bin` and then create the local keystore by executing the below `gsk8capicmd_64` command.

```
gsk8capicmd_64 -keydb -create -db "/home/<db2_instance>/sqllib/security/my-keystore.p12" -pw
"Temp1234" -type pkcs12 -stash
```

NOTE: Here the instance owner is db2inst2, Temp1234 is the password provided to the local keystore with –stash option with stores the keystore password in a stash file with the same base name as the keystore file but with the file extension ".sth".

The keystore and its stash file are created in the instance directory as follows:



Configuring DB2 Instance to Use a keystore

To configure a DB2 instance to use a keystore for native encryption, you just need to set two database manager configuration parameters: **keystore_type** and **keystore_location**.

For a local keystore, set keystore_type to "PKCS12", and set keystore_location to the absolute path and file name of the local keystore file.

```
db2 update dbm cfg using keystore_location /home/<db2_instance>/sqllib/security/my-keystore.p12
```

```
db2 update dbm cfg using keystore_type pkcs12
```

Restart the database server using `db2stop` and `db2start` command.

Creating Encrypted Database

With DB2 native encryption, when you create a database with the ENCRYPT parameter, by default the database manager creates a new master key for the database and adds that master key to the keystore.

To create an encrypted database with the default settings, specify the ENCRYPT option on the CREATE DATABASE command:

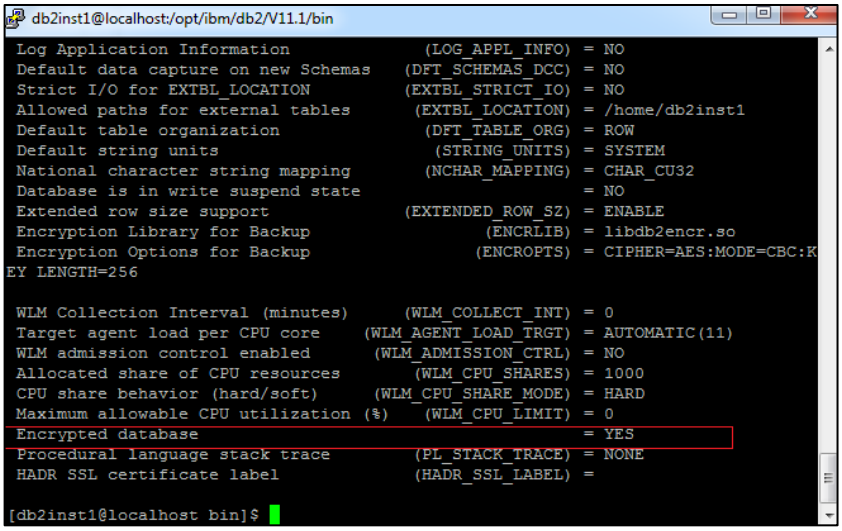
```
db2 create db myibm encrypt
```

NOTE: Here 'myibm' is the name of the encrypted database created.

Verifying a Database Native Encryption

To verify that your database has been successfully encrypted by DB2 native encryption, ensure that the value of the 'Encrypted database' db configuration parameter value is **YES** in the following command:

```
db2 get db cfg for myibm
```



Verifying that the software-based wallet is working fine

1. Connect to the database:

   ```
   db2 connect to myibm
   ```

2. Type `db2` press enter.

   ```
   bash-4.3$ db2
   (c) Copyright IBM Corporation 1993,2007
   Command Line Processor for DB2 Client 11.1.3.3

   You can issue database manager commands and SQL statements from the command
   prompt. For example:
       db2 => connect to sample
       db2 => bind sample.bnd

   For general help, type: ?.
   For command help, type: ? command, where command can be
   the first few keywords of a database manager command. For example:
    ? CATALOG DATABASE for help on the CATALOG DATABASE command
    ? CATALOG          for help on all of the CATALOG commands.

   To exit db2 interactive mode, type QUIT at the command prompt. Outside
   interactive mode, all commands must be prefixed with 'db2'.
   To list the current command option settings, type LIST COMMAND OPTIONS.

   For more detailed help, refer to the Online Reference Manual.

   db2 =>
   ```

3. Create a EMPLOYEE_SALARY table in the database:

   ```
   db2 => CREATE TABLE EMPLOYEE_SALARY (DEPTNO CHAR(3) NOT NULL,DEPTNAME VARCHAR(36) NOT NULL,EMPNO
   CHAR(6) NOT NULL,SALARY DECIMAL(9,2) NOT NULL WITH DEFAULT)
   ```

4. Enter some values in the EMPLOYEE_SALARY table:

   ```
   db2 => INSERT INTO EMPLOYEE_SALARY VALUES (001,'IT',001,10000)

   db2 => INSERT INTO EMPLOYEE_SALARY VALUES (001,'IT',002,15000)
   ```

5. Display the contents of the EMPLOYEE_SALARY table with the following command:

```
db2 => SELECT * FROM EMPLOYEE_SALARY
```

6. Verify access to keystore.

   Move or rename the encryption wallet to ensure that it is not available. Connect to the database, it should throw the error.

   Rename `my-keystore.p12` to `my-keystore.p24`

```
db2 => connect to myibm

SQL1728N  The command or operation failed because the keystore could not be

accessed. Reason code "2".
```

   Move the encryption wallet back to the keystore location for it to be accessible again.

Migrating from a local keystore to a PKCS #11 keystore

1. Create a PKCS#11 keystore.

   To store master keys in a PKCS #11 keystore with DB2 native encryption, you need to create a configuration file that contains details about the PKCS #11 keystore.

   On the DB2 server, create the PKCS #11 keystore configuration file luna.cfg with the following details:

```
VERSION=1
PRODUCT_NAME=Luna
ALLOW_KEY_INSERT_WITHOUT_KEYSTORE_BACKUP=true
LIBRARY=<LunaClient installation dir>/lib/libCryptoki2_64.so
SLOT_LABEL=<Partition_label>
NEW_OBJECT_TYPE=PRIVATE
KEYSTORE_STASH=/home/<db2_instance>/sqllib/security/pkcs11_pw.sth
```

   Here SLOT_LABEL identifies the slot in the HSM by a label. The label is a name that is defined by the application, and is assigned during token initialization.

   KEYSTORE_STASH is the absolute path and name of the stash file that holds the PKCS #11 keystore password. The instance uses the stash file to authenticate to the PKCS #11 keystore.

   Ensure that the ALLOW_KEY_INSERT_WITHOUT_KEYSTORE_BACKUP parameter is set to TRUE in the PKCS #11 keystore configuration file.

2. Create a stash file.

```
db2credman -stash -password <partition password> -to
/home/<db2_instance>/sqllib/security/pkcs11_pw.sth
```

   NOTE: It should be noted that storing the PKCS#11 keystore password in a stash file is optional. If the user wants to stash the password only then a stash file needs to be created and the path should be specified in 'KEYSTORE_STASH' in the configuration file.

3. Migrate the master key from the local keystore to the PKCS #11 keystore by issuing the **db2p12top11** command.

```
db2p12top11 -to /home/<db2_instance>/sqllib/security/luna.cfg -pin <partition password>
```

```
[db2inst2@localhost bin]$ db2p12top11 -to /home/db2inst2/sqllib/security/luna.cfg -pin temp123#
Migrating keys from <(null)> local keystore
to PKCS#11 HSM using vendor library </usr/safenet/lunaclient/lib/libcklog2.so>
defined in configuration file </home/db2inst2/sqllib/security/luna.cfg>.

Migrating key: <DB2_SYSGEN_db2inst2_MYIBM_2017-02-03-04.19.30_B9A0AA10> ... Successful.

Out of 1 key(s): 1 key(s) inserted successfully, 0 failed.
[db2inst2@localhost bin]$
```

On executing this command, the master key present on the local keystore should be migrated to the HSM partition which can be verified by 'partition showcontents' command on HSM.

```
Partition Name:                          test
Partition SN:                            1129959498113
Partition Label:                         test
Storage (Bytes): Total=95537, Used=176, Free=95361
Number objects:  1

Object Label:  DB2_SYSGEN_db2inst2_MYIBM_2017-02-03-04.19.30_B9A0AA10
Object Type:   Symmetric Key
Object Handle: 117
```

4. Set the ALLOW_KEY_INSERT_WITHOUT_KEYSTORE_BACKUP parameter to FALSE in the PKCS #11 keystore configuration file (luna.cfg).
5. Configure the DB2 instance to use the PKCS #11 keystore.

   ```
   db2 update dbm cfg using keystore_location /home/<db2_instance>/sqllib/security/luna.cfg
   ```

   ```
   db2 update dbm cfg using keystore_type pkcs11
   ```

Test if the database can reach the HSM device

1. Connect to the database.

   ```
   db2 connect to myibm
   ```

2. Display the contents of the EMPLOYEE table with the following command:

   ```
   db2 => SELECT * FROM EMPLOYEE_SALARY
   ```

   The user should be able to view the encrypted Table.

3. Rename the local keystore wallet to ensure that it is not available and the database is being accessed by the HSM master key. To verify that the database is being accessed by HSM wallet master key, run "`db2 get dbm cfg`" and the following output should appear:

   ```
   Keystore type                        (KEYSTORE_TYPE) = PKCS11
   ```

   ```
   Keystore location                    (KEYSTORE_LOCATION)
   =/home/<db2_instance>/sqllib/security/luna.cfg
   ```

4. Stop the NTLS service on HSM or break the NTLS connection and restart the database using `db2stop` and `db2start` commands.
5. Connect to the database should fail with the following error:

```
db2 => connect to myibm

SQL1783N  The command or operation failed because an error was encountered

accessing the PKCS #11 key manager.
```

Generating Master Encryption Key directly onto the HSM

1. Create a PKCS#11 keystore.

   On the DB2 server, create the PKCS #11 keystore configuration file luna.cfg:

   ```
   VERSION=1
   PRODUCT_NAME=Luna
   ALLOW_KEY_INSERT_WITHOUT_KEYSTORE_BACKUP=true
   LIBRARY=<LunaClient_Installation_Dir>/lib/ libCryptoki2_64.so
   SLOT_LABEL= <Partition_label>
   NEW_OBJECT_TYPE=PRIVATE
   KEYSTORE_STASH=/home/<db2_instance>/sqllib/security/pkcs11_pw.sth
   ```

   In the above configuration, SLOT_LABEL identifies the slot in the HSM by a label. The label is assigned during token initialization.

   KEYSTORE_STASH is the absolute path and name of the stash file that holds the PKCS #11 keystore password. The instance uses the stash file to authenticate to the PKCS #11 keystore.

   db2_instance is the instance owner name for integrating IBM db2 directly with the PKCS #11 keystore.

2. Create a stash file.

   ```
   db2credman -stash -password <partition password> -to /home/<db2
   instance>/sqllib/security/pkcs11_pw.sth
   ```

   NOTE: It should be noted that storing the PKCS#11 keystore password in a stash file is optional. If the user wants to stash the password only then a stash file needs to be created and the path should be specified in 'KEYSTORE_STASH' in the configuration file.

3. Configure a DB2 instance to use the PKCS #11 keystore.

   For a PKCS #11 keystore, set keystore_type to "PKCS11", and set keystore_location to the absolute path and PKCS #11 keystore configuration file luna.cfg.

   ```
   db2 update dbm cfg using keystore_location /home/<db2_instance>/sqllib/security/luna.cfg

   db2 update dbm cfg using keystore_type pkcs11
   ```

   These changes will take effect after the next `db2start` command. Restart the database server using `db2stop` and `db2start` command.

4. Create an encrypted database. To create an encrypted database with the default settings, specify the ENCRYPT option on the CREATE DATABASE command.

   ```
   db2 create db myencdb encrypt
   ```

   This command creates a new encrypted database and creates a master key in the HSM partition.

Verify that the HSM wallet is working fine:

1. Type db2 and press enter.

2. Connect to the database.

   ```
   db2 connect to myencdb
   ```

3. Create a STUDENT_MARKS table in the database.

   ```
   CREATE TABLE STUDENT_MARKS (CLASS_NO CHAR(3) NOT NULL,DEPTNAME VARCHAR(36)  NOT NULL,STUDNO
   CHAR(6) NOT NULL,MARKS CHAR(6) NOT NULL WITH DEFAULT)
   ```

4. Enter some values in the STUDENT_MARKS table.

   ```
   INSERT INTO STUDENT_MARKS VALUES (10,'SCIENCE',001,95)

   INSERT INTO STUDENT_MARKS VALUES (10,'COMMERCE',002,90)
   ```

5. Display the contents of the STUDENT_MARKS table with the following command:

   ```
   SELECT * FROM STUDENT_MARKS
   ```

6. Stop the NTLS service on HSM or break the NTLS connection and restart the database using `db2stop` and `db2start` commands.

   Connect to the database should fail with the following error:

   ```
   db2 => connect to myibm

   SQL1783N The command or operation failed because an error was encountered accessing the
   PKCS #11 key manager.
   ```

NOTE: To encrypt an already existing unencrypted database, the master key created while creating an encrypted database can be used. To encrypt an already existing database refer step 3 in Appendix.

3
Appendix

1. The installation requirements for data server products are listed in: http://www.ibm.com/support/knowledgecenter/en/SSEPGG_11.1.0/com.ibm.db2.luw.qb.server.doc/doc/r0008865.html
2. The installation procedure of IBM DB2 is described in: https://www.ibm.com/support/knowledgecenter/SSEPGG_11.1.0/com.ibm.swg.im.dbclient.install.doc/doc/t0007317.html
3. Steps to Encrypting an existing unencrypted database:
   a. Generate a backup image of the database you would like to encrypt:

      ```
      db2 deactivate db diribm

      db2 backup db diribm to /tmp/
      ```

   b. Drop the original copy of the database you wanted to encrypt:

      ```
      db2 drop db diribm
      ```

   c. Restore the backup image into a new encrypted database :

      ```
      db2 restore database DIRIBM into mynewdb encrypt master key label myMK
      ```

NOTE: Here ‘myMK’ is the label of the master key present on the HSM partition.