SAP

Sybase ASE

16.1

Integration Guide

u.trust GP HSM Se-Series

SecurityServer 6.2.0

utimaco®

Imprint

Copyright 2026	Utimaco IS GmbH
	Krefelder Straße 220
	52070 Aachen
	Germany
Phone	AMERICAS +1-844-UTIMACO (+1 844-884-6226)
	EMEA +49 800-627-3081
	APAC +81 800-919-1301
Internet	https://support.hsm.utimaco.com/
e-mail	support@utimaco.com
Document Version	1.0.0
Date	2026-04-10
Status	PUBLISHED
Document No.	IG-2026-0035

Table of Contents

1 Introduction

1.1 About This Guide

This guide provides detailed instructions on integrating SAP Sybase ASE with the Utimaco SecurityServer Hardware Security Module (HSM). It describes how to configure the database environment to securely generate, store, and operate cryptographic keys inside the HSM using the Utimaco PKCS#11 interface. The guide outlines the installation prerequisites, configuration procedures, security considerations, and verification steps required to ensure a secure and fully functional integration environment.

1.2 Target Audience

This guide is intended for database administrators of SAP Sybase ASE and administrators of Utimaco HSMs.

1.3 Purpose of the Integration

The purpose of this integration is to enable SAP Sybase ASE to enhances the security and compliance of data protection by leveraging the Utimaco SecurityServer HSM as the trusted hardware-based key management and encryption backend. By storing all sensitive cryptographic keys inside the HSM and performing operations such as key generation, encryption, decryption, and signing within its protected boundary, the integration enhances database security, ensures strong key protection, and supports regulatory and compliance requirements. This approach eliminates exposure of keys on the application host, strengthens data-at-rest and data-in-use security for Sybase ASE environments, and provides centralized, tamper-resistant key lifecycle management to meet enterprise-grade security expectations.

1.4 Abbreviations

Abbreviation	Meaning
HSM	Hardware Security Module
ASE	Adaptive Server Enterprise

Abbreviation	Meaning
PKI	Public Key Infrastructure
PKCS	Public Key Cryptography Standards
PKCS#11	PKCS Part 11: The Cryptographic Token Interface Standard
SO	The PKCS#11 cryptographic slot Security Officer
DB	Database
JRE	Java Runtime Environment
MBK	Master backup key
P11CAT	the PKCS#11 graphical interface tool
FIPS	Federal Information Processing Standards
SAP	Systems, Applications, and Products in Data Processing
TDE	Transparent Data Encryption
DEK	Database Encryption Key
CU	Crypto User
ISQL	Interactive SQL Utility
OLTP	Online Transaction Processing

Abbreviation	Meaning
LAN	Local Area Network
PCIe	Peripheral Component Interconnect Express
TLS	Transport Layer Security
SSL	Secure Sockets Layer
OS	Operating System

Table 1: Abbreviations

1.5 Document Conventions

The following conventions are used in this guide:

Convention	Use	Example
Bold	Items of the Graphical User Interface (GUI), e.g., menu options	Press **OK**
`Monospaced`	Code that is given for explanation or as an example, file paths	`sp_encryption helpkey`
Italic	References and important terms	See *Sample Chapter* in the *CryptoServer - Sample Manual*

Table 2: Document Conventions

We use special icons to highlight the most important notes and information.

Here you will find important safety information that should be followed.

Here you will find additional notes or supplementary information.

This message indicates the expected result after the successful execution of an instruction.

2 Product Overview

2.1 Overview of SAP Sybase ASE

SAP Sybase Adaptive Server Enterprise (ASE) is a high-performance relational database server optimized for mission-critical online transaction processing (OLTP). It provides fast, reliable, and scalable transaction execution, supporting large data volumes and high user concurrency for industries such as finance, telecom, and healthcare. ASE includes capabilities for workload optimization, high availability, disaster recovery, encryption, and a graphical administration cockpit for monitoring and management, making it a robust choice for enterprise-grade data processing environments.

2.2 Overview of Utimaco GP HSM

u.trust GP HSM Se-Series is a hardware security module developed by Utimaco IS GmbH. It is a physically protected, specialized computer unit designed to perform sensitive cryptographic tasks and securely manage and store cryptographic keys and data. It can be used as a universal, independent security component for heterogeneous computer systems.

2.3 Joint Value Proposition

Integrating SAP Sybase ASE with Utimaco Hardware Security Modules (HSMs) delivers a strong, hardware-anchored security foundation for protecting all cryptographic keys used for database encryption, secure communication, authentication, and regulatory compliance. Utimaco HSMs provide tamper-resistant, FIPS-certified key generation and storage, ensuring that sensitive keys never leave secure hardware boundaries and safeguarding ASE's encrypted data, column-level protection, SSL/TLS communication, and advanced security controls. This integration helps enterprises meet stringent regulatory requirements across industries.

By offloading cryptographic operations to dedicated, high-performance Utimaco u.trust GP HSM Se-Series hardware, organizations benefit from improved performance, reduced risk, and a scalable architecture aligned with ASE's high-throughput transactional engine. Utimaco's support for standard interfaces (such as PKCS#11) enables seamless integration into ASE environments without application changes, while features such as secure key backup and remote management ensure operational continuity. Together, SAP Sybase ASE and u.trust GP HSM Se-Series provide a secure, compliant, and scalable database platform suitable for modern business-critical workloads.

3 Integration Requirements and Prerequisites

3.1 Tested Versions

SAP Sybase ASE	Utimaco Security Server Version	Utimaco HSM
v16.1	SecurityServer 6.2.0 p11tool2 from Utimaco SecurityServer product package	SecurityServer CSe-Series/Se-Series

Table 3: Tested Versions

3.2 Hardware and Software Requirements

Software Requirements:

Software	Software Requirements
SAP Sybase ASE	v16.1
HSM Utility	SecurityServer PKCS#11 Tool (p11tool2) v6.2.0
HSM Interfaces	SecurityServer PKCS#11 Provider v6.2.0
Operating System	Rocky Linux v9.6

Table 4: Software Requirements

Hardware Requirements:

Hardware	Hardware Requirements
Utimaco LAN HSM	u.trust GP HSM Se-Series LAN with SecurityServer 6.2.0 firmware or higher

Hardware	Hardware Requirements
Utimaco PCI-e HSM	u.trust GP HSM Se-Series PCI-e with SecurityServer 6.2.0 firmware or higher

Table 5: Hardware Requirements

3.3 Prerequisites

- SecurityServer and SAP Sybase ASE are installed and configured according to the Tested Versions.
- A SecurityServer administrator user is available with the required privileges.
- A valid SAP Sybase ASE license is installed and active.

4 Installation and Configuration

4.1 Setting Up Utimaco Security Server

4.1.1 Download and Install the Utimaco Software

If you have not already done so, create and request an Utimaco Support Portal Account at https://support.hsm.utimaco.com/support. This will allow you to download the software components needed for this installation.

Log in to the Utimaco Support Portal and download the u.trust GP HSM Se-Series package: https://support.hsm.utimaco.com/support/downloads/u.trust-anchor-se-series.

4.2 Setting Up Sybase ASE

4.2.1 Download and Install Sybase ASE

Please refer to the below link for downloading and installing SAP Sybase ASE in Linux:

https://help.sap.com/docs/SAP_ASE/23c3bb4a29be443ea887fa10871a30f8/a6612e5fbc2b10149d8a80b52f34dc5a.html?q=showserver

The Transparent Data Encryption (TDE) option must be enabled during the SAP Sybase ASE installation.

In this integration, the `/opt/sap` directory is used to install SAP Sybase ASE, and the same path is referenced in all subsequent steps. Users may choose a different installation directory during setup; however, the selected path must be used consistently in all further configuration and integration steps.

Verify the SAP Sybase database version using the command below:

```
[sap@master-node ~]$ isql -SSAP01 -Usa
Password:
1> SELECT @@VERSION
2> go
1> SELECT @@SERVERNAME
```

```
2> GO
```

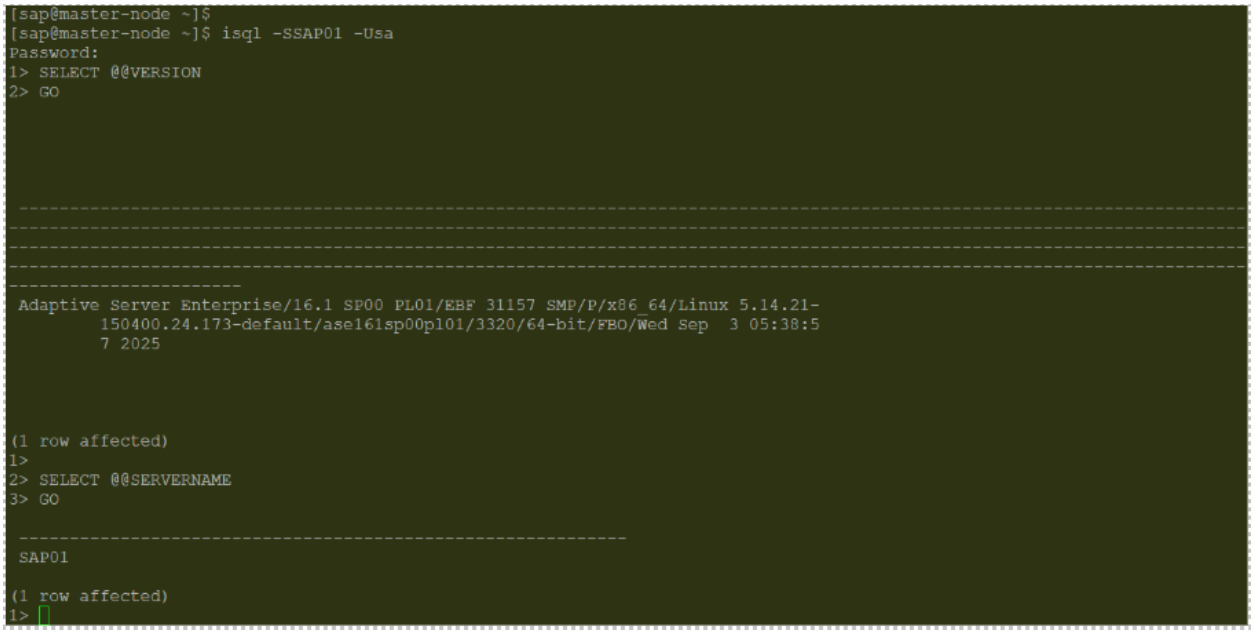

Figure 1 : Version Information of SAP Sybase ASE Server

The ASE server can be accessed using the `isql -S<Server name> -Usa` command.

5 Integration Steps

SAP Sybase ASE supports Transparent Data Encryption for protecting data at rest. The key protection using an External Keystore (HSM) is achieved through an envelope encryption model. The database is encrypted using a Database Encryption Key (DEK), which performs the actual data encryption. This DEK is protected by the ASE master key, which acts as a Key Encryption Key (KEK) within the database. The ASE master key itself is then protected by an external key stored in the HSM via PKCS#11.

5.1 Configuration for Utimaco Security Server

5.1.1 Initialize a Slot

1. Initialize a slot with a custom label using the `p11tool2`.
2. Create the SO or Security Officer using p11tool2. Then, using the `p11tool2` command, initialize the slot that you want to use and the slot user, as shown below:

```
$ ./p11tool2 slot=<slot_no> Label=<token_label> Login=ADMIN,ADMIN.key
InitToken=<SO_PIN>
$ ./p11tool2 slot=<slot_no> LoginSO=<SO_PIN> InitPin=<CryptoUser_PIN>
```

5.2 Configuration on Sybase ASE

1. Copy the PKCS#11 library `libcs_pkcs11_R3.so` to the SAP Sybase ASE library folder `/opt/sap/ASE-16_1/lib/`.
2. Copy the PKCS#11 config file `cs_pkcs11_R3.cfg` to the SAP Sybase ASE home library `/opt/sap/config/`.
3. Update the PKCS#11 config file `cs_pkcs11_R3.cfg`.
4. Update the environment variables in `SYBASE.sh` for the u.trust GP HSM Se-Series. Add the environment variable for PKCS#11 R3 library and config file.

```
[sap@master-node ~]$
[sap@master-node ~]$ diff -Nurb base/SYBASE.sh modification/SYBASE.sh
--- base/SYBASE.sh      2026-03-30 09:57:27.465045284 -0700
+++ modification/SYBASE.sh      2026-03-30 01:45:56.667953529 -0700
@@ -16,7 +16,7 @@
 #
 # Replace lib, lib3p, and lib3p64 with devlib, devlib3p, and devlib3p64 when debugging
 #
-LD_LIBRARY_PATH="/opt/sap/OCS-16_1/lib:/opt/sap/OCS-16_1/lib3p64:/opt/sap/OCS-16_1/lib3p":$LD_LIBRARY_PATH
+LD_LIBRARY_PATH="/opt/sap/OCS-16_1/lib:/opt/sap/OCS-16_1/lib3p64:/opt/sap/OCS-16_1/lib3p:/opt/sap/ASE-16_1/lib":$LD_LIBRARY_PATH
 export LD_LIBRARY_PATH
 LD_LIBRARY_PATH="/opt/sap/DataAccess/ODBC/lib":$LD_LIBRARY_PATH
 export LD_LIBRARY_PATH
@@ -38,3 +38,5 @@
 export SYBASE_WS
 PATH="/opt/sap/ASE-16_1/jobscheduler/bin":$PATH
 export PATH
+CS_PKCS11_R3_CFG="/opt/sap/config/cs_pkcs11_R3.cfg"
+export CS_PKCS11_R3_CFG
[sap@master-node ~]$
```

Figure 2 : Add Environment Variables in SYBASE.sh

5. Include the SYBASE.sh execution in '.bash_profile'.

```
# Load SAP Sybase ASE environment automatically

if [ -f "$HOME/SYBASE.sh" ]; then
    . "$HOME/SYBASE.sh"
fi
```

6. Update the environment variables in RUN_SAP01 and RUN_SAP01_BS for the u.trust GP HSM Se-Series. Both files should have same changes as per the below screenshot. Both files are available in `/ASE-16_1/install/RUN_SAP01` and `/ASE-16_1/install/RUN_SAP01/RUN_SAP01_BS`.

```
[sap@master-node ~]$
[sap@master-node ~]$ diff -Nurb base/RUN_SAP01 modification/RUN_SAP01
--- base/RUN_SAP01      2026-03-30 01:44:43.418496518 -0700
+++ modification/RUN_SAP01      2026-03-30 01:31:21.478443592 -0700
@@ -11,6 +11,13 @@
 #


+# Path to PKCS#11 config required by ASE TDE
+export CS_PKCS11_R3_CFG="/opt/sap/config/cs_pkcs11_R3.cfg"
+
+
+# Ensure ASE can find libcs_pkcs11_R3.so
+export LD_LIBRARY_PATH="/opt/sap/ASE-16_1/lib:${LD_LIBRARY_PATH}"
+
 /opt/sap/ASE-16_1/bin/dataserver \
 -d/opt/sap/data/master.dat \
 -e/opt/sap/ASE-16_1/install/SAP01.log \
[sap@master-node ~]$
```

Figure 3 : RUN_SAP01 Changes

7. Shutdown the ASE server then restart the ASE server using `startserver`.

```
1> shutdown
2> go
```

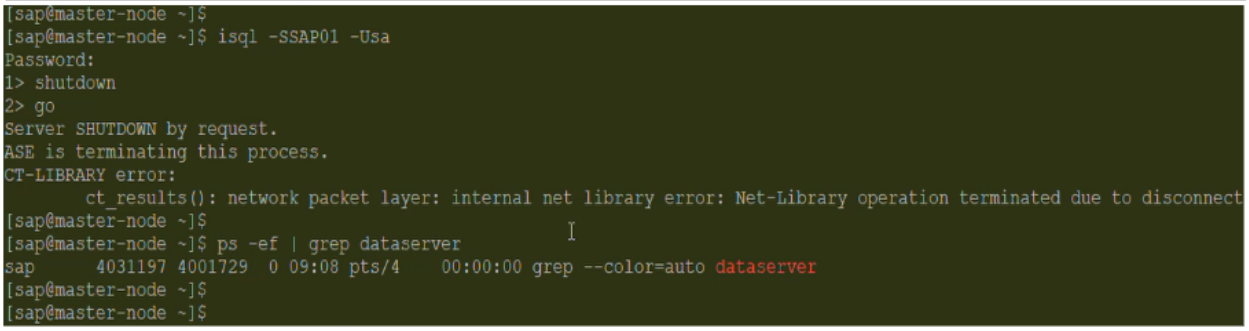

Figure 4 : ASE Server Shutdown

```
$cd ASE-16_1/install/
$/opt/sap/ASE-16_1/bin/startserver -f RUN_SAP01
```

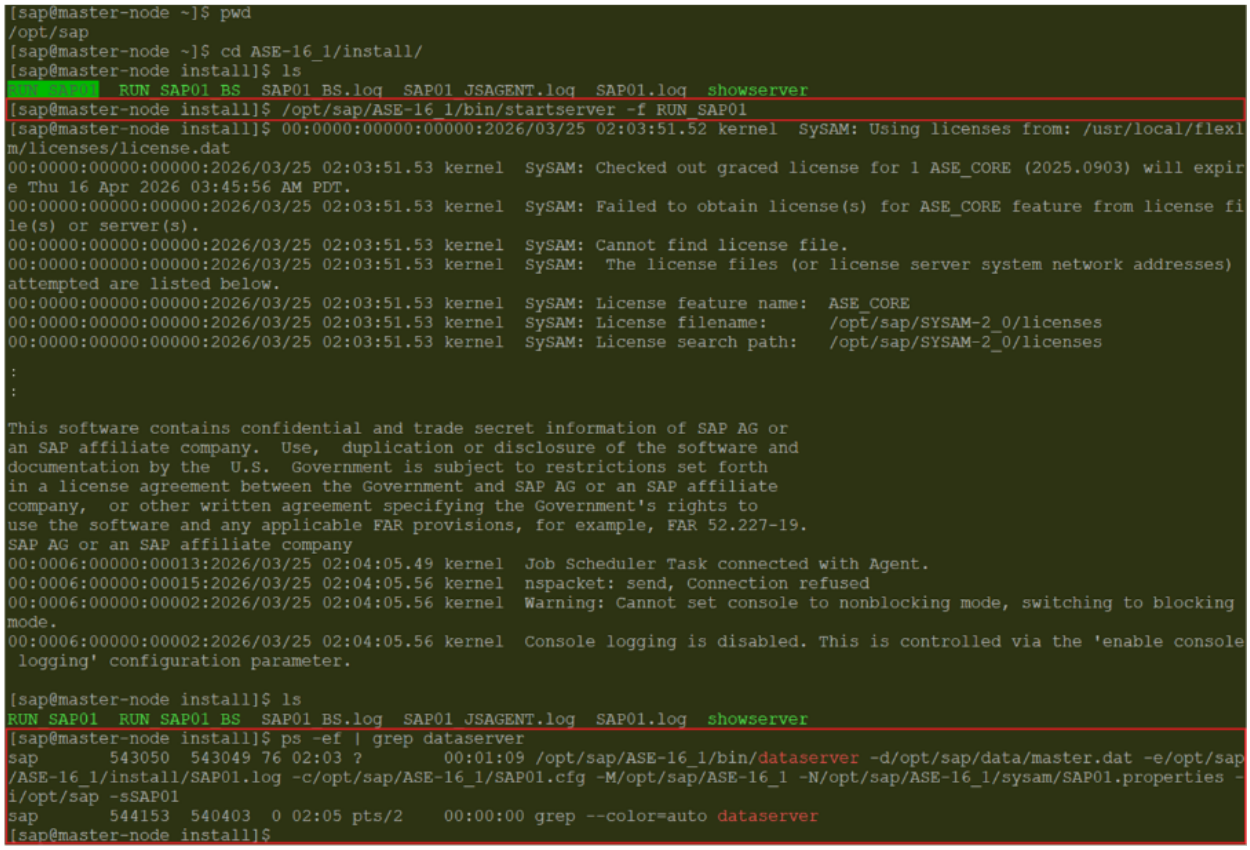

Figure 5 : Start ASE Server

5.2.1 Setting Environment for Utimaco Security Server Access

Setting environment variables for Utimaco Security Server via PKCS11:

1. Copy the PKCS#11 library `libcs_pkcs11_R3.so` to the ASE library path `/opt/sap/ASE-16_1/lib/`.
2. Copy the PKCS#11 config file `cs_pkcs11_R3.cfg` to the `config` folder in SAP home directory `/opt/sap/config`.
3. Update the PKCS#11 config file `cs_pkcs11_R3.cfg`.

```
[Global]
# For Unix:
#Logpath = /tmp
# For Windows:
 Logpath = C:/ProgramData/Utimaco/PKCS11_R3
```

```
# Loglevel (0 = NONE; 1 = ERROR; 2 = WARNING; 3 = INFO; 4 = TRACE)
Logging = 1
# Prevents expiring session after inactivity of 15 minutes
KeepAlive = true
# Set the Device to connect with
#[CryptoServer]
# Device specifier
Device = <port>@<HSM_IP>
```

Replace `Device` parameter with HSM device IP and port details

For detailed guidance on commands and their parameters, please refer to the Utimaco CryptoServer documentation. The device could be a CryptoServer HSM, available in either PCIe or LAN form factors. Depending on the type, the device configuration line will follow one of these formats:

LAN-based HSM:

Device = 288@ipaddress

PCIe-based HSM:

Device = /dev/cs2.0

Be sure to select the appropriate format based on your specific hardware setup.

To simplify your testing process, it's recommended that you enable the PKCS#11 log file by adjusting the logging settings.

Specifically: Set the LogPath to a writable directory (not a specific file). Set the Logging Loglevel to 1 for basic logging. Increase it to 4 for more detailed output during testing. This will generate a log file named `cs_pkcs11_R3.log` within the specified LogPath directory. Reviewing this log can help with troubleshooting if you encounter issues.

Once the testing is complete, it's advisable to reduce Logging Loglevel to limit the output to only critical or important messages

4. Update the environment variables in `SYBASE.sh` for the u.trust GP HSM Se-Series. Add the environment variable for the PKCS#11 R3 library and config file:

```
[sap@master-node ~]$
[sap@master-node ~]$ diff -Nurb base/SYBASE.sh modification/SYBASE.sh
--- base/SYBASE.sh      2026-03-30 09:57:27.465045284 -0700
+++ modification/SYBASE.sh      2026-03-30 01:45:56.667953529 -0700
@@ -16,7 +16,7 @@
 #
 # Replace lib, lib3p, and lib3p64 with devlib, devlib3p, and devlib3p64 when debugging
 #
-LD_LIBRARY_PATH="/opt/sap/OCS-16_1/lib:/opt/sap/OCS-16_1/lib3p64:/opt/sap/OCS-16_1/lib3p":$LD_LIBRARY_PATH
+LD_LIBRARY_PATH="/opt/sap/OCS-16_1/lib:/opt/sap/OCS-16_1/lib3p64:/opt/sap/OCS-16_1/lib3p:/opt/sap/ASE-16_1/lib":$LD_LIBRARY_PATH
 export LD_LIBRARY_PATH
 LD_LIBRARY_PATH="/opt/sap/DataAccess/ODBC/lib":$LD_LIBRARY_PATH
 export LD_LIBRARY_PATH
@@ -38,3 +38,5 @@
 export SYBASE_WS
 PATH="/opt/sap/ASE-16_1/jobscheduler/bin":$PATH
 export PATH
+CS_PKCS11_R3_CFG="/opt/sap/config/cs_pkcs11_R3.cfg"
+export CS_PKCS11_R3_CFG
[sap@master-node ~]$
```

Figure 6 : Environment Variable Details for SYBASE.sh

5. Include the `SYBASE.sh` execution in '.bash_profile':

```
# Load SAP Sybase ASE environment automatically

if [ -f "$HOME/SYBASE.sh" ]; then
    . "$HOME/SYBASE.sh"
fi
```

6. Update the environment variables in `RUN_SAP01` and `RUN_SAP01_BS` for the u.trust GP HSM Se-Series. Both files are available in `/ASE-16_1/install/RUN_SAP01` and `/ASE-16_1/install/RUN_SAP01/RUN_SAP01_BS`.

```
+# Path to PKCS#11 config required by ASE TDE
+export CS_PKCS11_R3_CFG="/opt/sap/config/cs_pkcs11_R3.cfg"
+# Ensure ASE can find libcs_pkcs11_R3.so
+export LD_LIBRARY_PATH="/opt/sap/ASE-16_1/lib:${LD_LIBRARY_PATH}"
+
  /opt/sap/ASE-16_1/bin/datasever \
  -d/opt/sap/data/master.dat \
  -e/opt/sap/ASE-16_1/install/SAP01.log \
```

7. Shutdown the ASE server then restart the ASE server using `startserver`:

```
1>shutdown
2>go
```

```
[sap@master-node ~]$
[sap@master-node ~]$ isql -SSAP01 -Usa
Password:
1> shutdown
2> go
Server SHUTDOWN by request.
ASE is terminating this process.
CT-LIBRARY error:
       ct_results(): network packet layer: internal net library error: Net-Library operation terminated due to disconnect
[sap@master-node ~]$
[sap@master-node ~]$ ps -ef | grep dataserver
sap      4031197 4001729  0 09:08 pts/4    00:00:00 grep --color=auto dataserver
[sap@master-node ~]$
[sap@master-node ~]$
```

Figure 7 : Shutdown from ASE server

```
#cd ASE-16_1/install/
#/opt/sap/ASE-16_1/bin/startserver -f RUN_SAP01
```

```
[sap@master-node ~]$ pwd
/opt/sap
[sap@master-node ~]$ cd ASE-16_1/install/
[sap@master-node install]$ ls
RUN_SAP01  RUN_SAP01_BS  SAP01_BS.log  SAP01_JSAGENT.log  SAP01.log  showserver
[sap@master-node install]$ /opt/sap/ASE-16_1/bin/startserver -f RUN_SAP01
[sap@master-node install]$ 00:0000:00000:00000:2026/03/25 02:03:51.52 kernel  SySAM: Using licenses from: /usr/local/flexl
m/licenses/license.dat
00:0000:00000:00000:2026/03/25 02:03:51.53 kernel  SySAM: Checked out graced license for 1 ASE_CORE (2025.0903) will expir
e Thu 16 Apr 2026 03:45:56 AM PDT.
00:0000:00000:00000:2026/03/25 02:03:51.53 kernel  SySAM: Failed to obtain license(s) for ASE_CORE feature from license fi
le(s) or server(s).
00:0000:00000:00000:2026/03/25 02:03:51.53 kernel  SySAM: Cannot find license file.
00:0000:00000:00000:2026/03/25 02:03:51.53 kernel  SySAM:  The license files (or license server system network addresses)
attempted are listed below.
00:0000:00000:00000:2026/03/25 02:03:51.53 kernel  SySAM: License feature name:  ASE_CORE
00:0000:00000:00000:2026/03/25 02:03:51.53 kernel  SySAM: License filename:      /opt/sap/SYSAM-2_0/licenses
00:0000:00000:00000:2026/03/25 02:03:51.53 kernel  SySAM: License search path:   /opt/sap/SYSAM-2_0/licenses

:
:

This software contains confidential and trade secret information of SAP AG or
an SAP affiliate company.  Use,  duplication or disclosure of the software and
documentation by the  U.S.  Government is subject to restrictions set forth
in a license agreement between the Government and SAP AG or an SAP affiliate
company,  or other written agreement specifying the Government's rights to
use the software and any applicable FAR provisions, for example, FAR 52.227-19.
SAP AG or an SAP affiliate company
00:0006:00000:00013:2026/03/25 02:04:05.49 kernel  Job Scheduler Task connected with Agent.
00:0006:00000:00015:2026/03/25 02:04:05.56 kernel  nspacket: send, Connection refused
00:0006:00000:00002:2026/03/25 02:04:05.56 kernel  Warning: Cannot set console to nonblocking mode, switching to blocking
mode.
00:0006:00000:00002:2026/03/25 02:04:05.56 kernel  Console logging is disabled. This is controlled via the 'enable console
 logging' configuration parameter.

[sap@master-node install]$ ls
RUN_SAP01  RUN_SAP01_BS  SAP01_BS.log  SAP01_JSAGENT.log  SAP01.log  showserver
[sap@master-node install]$ ps -ef | grep dataserver
sap       543050  543049 76 02:03 ?        00:01:09 /opt/sap/ASE-16_1/bin/dataserver -d/opt/sap/data/master.dat -e/opt/sap
/ASE-16_1/install/SAP01.log -c/opt/sap/ASE-16_1/SAP01.cfg -M/opt/sap/ASE-16_1 -N/opt/sap/ASE-16_1/sysam/SAP01.properties -
i/opt/sap -sSAP01
sap       544153  540403  0 02:05 pts/2    00:00:00 grep --color=auto dataserver
[sap@master-node install]$
```

Figure 8 : Start ASE Server

5.2.2 Enable ASE Encryption Feature

Configure SAP ASE for Full Database Encryption:

```
1>sp_configure 'enable encrypted columns',1
2>go
```

```
[sap@master-node ~]$ isql -SSAP01 -Usa
Password:
1> sp_configure 'enable encrypted columns',1
2> go
 Parameter Name                                  Default
        Memory Used               Config Value
        Run Value                 Unit         Type
 ------------------------------------------- ---------------------
        ------------------------ --------------------------
        ----------------------- ----------- -------------
 enable encrypted columns                                      0
                4108                          1
                   1            switch       dynamic

(1 row affected)
Configuration option changed. ASE need not be rebooted since the option is
dynamic.
Changing the value of 'enable encrypted columns' to '1' increases the amount of
memory ASE uses by 2 K.
(return status = 0)
```

Figure 9 : Enable ASE Encryption

5.2.3 Enable HSM as External Keystore and Update HSM credentials in the ASE server

1. Enable HSM as External Keystore:

Enable SAP ASE to store encryption keys externally by specifying the `external keystore` configuration parameter.

```
1>sp_configure 'external keystore', 0, 'hsm'
2>go
```

```
1> sp_configure 'external keystore', 0, 'hsm'
2> go
 Parameter Name                    Default
        Memory Used                Config Value
        Run Value                  Unit     Type
 ---------------------------------- ---------------------
        ------------------------ --------------------------
        ----------------------- -------- -------------
 external keystore                        NULL
                 0                          hsm
               hsm             name       dynamic

(1 row affected)
Resulting configuration value and memory use have not changed from previous
values: new configuration value hsm, previous value hsm.
(return status = 0)
```

Figure 10 : Enable HSM as External Keystore

2. Update HSM credentials in the ASE server:

The HSM device uses the PKCS#11 library to define a platform-independent Cryptoki API to cryptographic tokens. SAP ASE requires authentication by the crypto user to create Cryptoki objects in the HSM device and perform encryption and decryption. Use the `sp_encryption` system procedure to specify the crypto user credentials to SAP ASE.

```
1>sp_encryption 'hsm_credential',
2>'lib=libcs_pkcs11_R3.so; pin=87654321; slot=0'
3>go
```

```
1> sp_encryption 'hsm_credential',
2> 'lib=libcs_pkcs11_R3.so; pin=87654321; slot=0'
3> go
(return status = 0)
1>
```

Figure 11 : Updated HSM credentials in the ASE server

5.2.4 Create HSM Key and a Master Key

1. Create a HSM Key:

Use the `create encryption key` command and the credentials set by `sp_encryption` to create the HSM key. SAP ASE creates the HSM key in the master database and uses the HSM key to encrypt only the master key of any database. SAP ASE supports only one HSM key per instance.

```
1>create encryption key hsm_key on external keystore
2>with keylength 256 init_vector random
3>go
```

```
1> sp_encryption 'hsm_credential',
2> 'lib=libcs_pkcs11_R3.so; pin=87654321; slot=0'
3> go
(return status = 0)
1> create encryption key hsm_key on external keystore
2> with keylength 256 init_vector random
3> go
1>
```

Figure 12 : Create HSM Key

2. Verify HSM key generation. The new HSM key `hsm_key` should be generated in the master:

```
1> sp_encryption helpkey
2> go
```

```
1> sp_encryption helpkey
2> go
 Key Name
        Key Owner
        Key Length
        Key Algorithm
        Key Type                                Pad
        Initialization Vector
        Protected By
        Key Recovery
        # of Key Copies
 ------------------------------------
        ---------------------------------------
        ----------------------------------------
        ------------------------------------------------
        ------------------------------------ ----------------------------
        ------------------------------------------------------------------
        ------------------------------------------------
        ------------------------------------------------
        --------------------------------------------------
 hsm_key
        dbo
                        256
        AES
        symmetric hsm key                                       0
                                    1
        hsm
                                0

 syb_hsmcredentialkey
        dbo
                        256
        AES
        hsm credential                                          0
                                    0
        static key
                                0
                                   0

(2 rows affected)
(return status = 0)
```

Figure 13 : HSM Key Generated

3. Verify the HSM key details:

```
1> sp_encryption helpkey,hsm_key
2> go
```

```
1> sp_encryption helpkey,hsm_key
2> go
 Key Name              Key Owner
        Key Length
        Key Algorithm
        Key Type                              Pad
        Initialization Vector
        Protected By
        Key Recovery
        # of Key Copies
        UUID
 --------------------- ----------------------------------------
        --------------------------------------
        ----------------------------------------------
        ------------------------------------ ---------------------------
        ------------------------------------------------------------
        ----------------------------------------------
        ----------------------------------------------
        ------------------------------------------------------
        ------------------------------------------------------------
 hsm_key               dbo
                          256
        AES
        symmetric hsm key                                    0
                                      1
        hsm
                               0
                                  0
        cbc77a633b70114e64a4b9db1c1d8292

(1 row affected)
(return status = 0)
1>
```

Figure 14 : HSM Key Details

4. Verify the key got updated in HSM. The key ID of the key in ASE server and key ID of the key in HSM should be same.

```
$./p11tool2 LoginUser=<crypto user PIN> ListObjects
```

```
[admin@master-node Administration]$
[admin@master-node Administration]$ ./p11tool2 LoginUser=87654321 ListObjects


CKO_SECRET_KEY:

+ 1.1
  CKA_KEY_TYPE                  = CKK_AES
  CKA_UNIQUE_ID                 = DB2E6136-202F-46BD-9EB3-C1D880BDCFC0
  CKA_SENSITIVE                 = CK_TRUE
  CKA_EXTRACTABLE               = CK_FALSE
  CKA_LABEL                     = SAP ASE instance key
  CKA_ID                        =
                0x63626337 37613633 33623730 31313465 |cbc77a633b70114e|
                  36346134 62396462 31633164 38323932 |64a4b9db1c1d8292|

[admin@master-node Administration]$
```

Figure 15 : Created HSM Key Displayed in HSM

5. Create the master key protected by the HSM:

Use the `create encryption key` command with the `master` parameter to create a master key encrypted with the HSM key.

```
1> create encryption key master with external key
2> go
1> sp_encryption helpkey
2> go
```

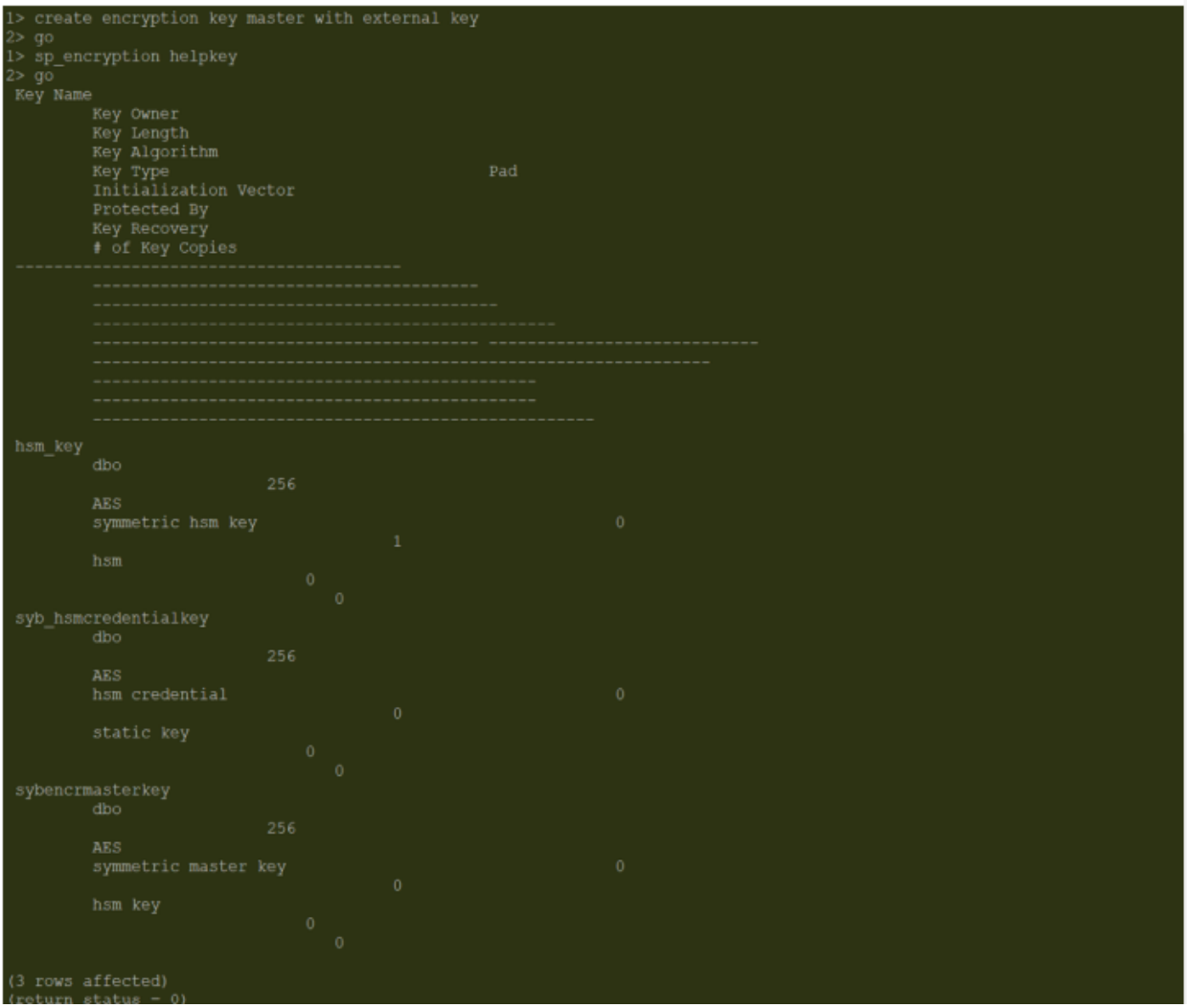

Figure 16 : Master Key Details

6 Verification and Testing

6.1 Functional Testing

To verify the integration, a test database 'tde_test' is created. Data is inserted into this database a Database Encryption Key (DEK) is used for encrypting the database. Enable encryption for the database and verify the database is accessible.
By validating data accessibility across ASE restarts ensures encryption remains intact, verifies correct key hierarchy enforcement and HSM connectivity via PKCS#11.

6.1.1 Create a Database Encryption Key (DEK)

1. Check the existing database:

```
1>SELECT name FROM sysdatabases
2>go
```

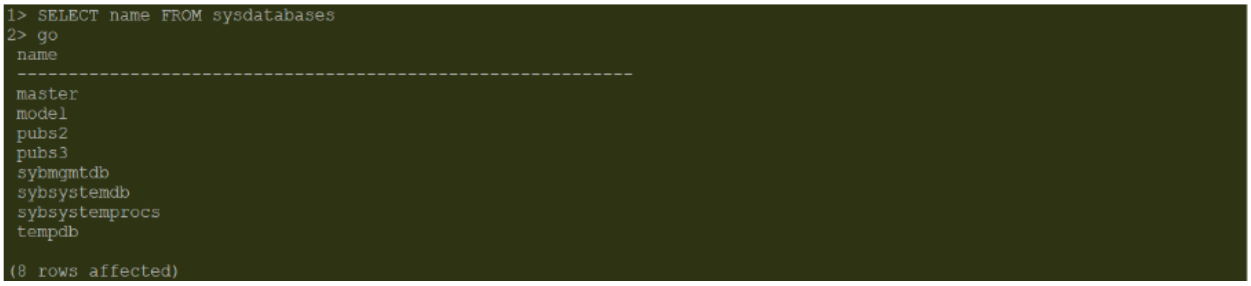

Figure 17 : Existing Database Details

2. Create a database:

```
1>create database tde_test
2>go
```

```
1> create database tde_test
2> go
CREATE DATABASE: allocating 1536 logical pages (6.0 megabytes) on disk 'master'
(1536 logical pages requested).
Database 'tde_test' is now online.
Database 'tde_test' has been created with success.
1> SELECT name FROM sysdatabases
2> go
 name
 ------------------------------------------------------------
 master
 model
 pubs2
 pubs3
 sybmgmtdb
 sybsystemdb
 sybsystemprocs
 tde_test
 tempdb

(9 rows affected)
```

Figure 18 : Create a Database

3. Create a table and insert data:

```
1>use tde_test
2>go

1>>create table emp_details (   emp_id  int  not null,
2>>emp_name     varchar(100),
3>>designation  varchar(50) )
4>>go

1>alter table emp_details
2>add constraint pk_emp_details primary key (emp_id)
3>go

1>insert into emp_details   values (1, 'Alex', 'Engineer')
2>go
1>insert into emp_details values (2, 'John', 'Finance')
2>go
1>insert into emp_details values (3, 'Mark', 'Manager')
2>go
```

```
1> use tde_test
2> go
1> create table emp_details (
2>    emp_id      int          not null,
3>    emp_name    varchar(100),
4>    designation varchar(50)
5> )
6> go
1> alter table emp_details
2> add constraint pk_emp_details primary key (emp_id)
3> go
1> insert into emp_details
2> values (1, 'Alex', 'Engineer')
3> go
(1 row affected)
1> insert into emp_details values
2> (2, 'John', 'Finance')
3> go
(1 row affected)
1> insert into emp_details values
2> (3, 'Mark', 'Manager')
3> go
(1 row affected)
```

Figure 19 : Table created and Data inserted

4. Verify the table details:

```
1>select * from emp_details
2>go
```

```
1> select * from emp_details
2> go
 emp_id
        emp_name

        designation
 -----------
        ----------------------------------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------
        ------------------------------------------------------------------------------------------------------------
          1
        Alex

        Engineer
          2
        John

        Finance
          3
        Mark

        Manager

(3 rows affected)
```

Figure 20 : Table Details

5. Create a Database Encryption Key (DEK):

The database encryption key (DEK) is created in the master database and used to encrypt a database. Use the `create encryption key` command in the `master` database to create a database encryption key. The DEK used here is 'tde_key'.

```
1>use master
2>go

1>create encryption key tde_key for database encryption
2>go

1>sp_encryption helpkey
2>go
```

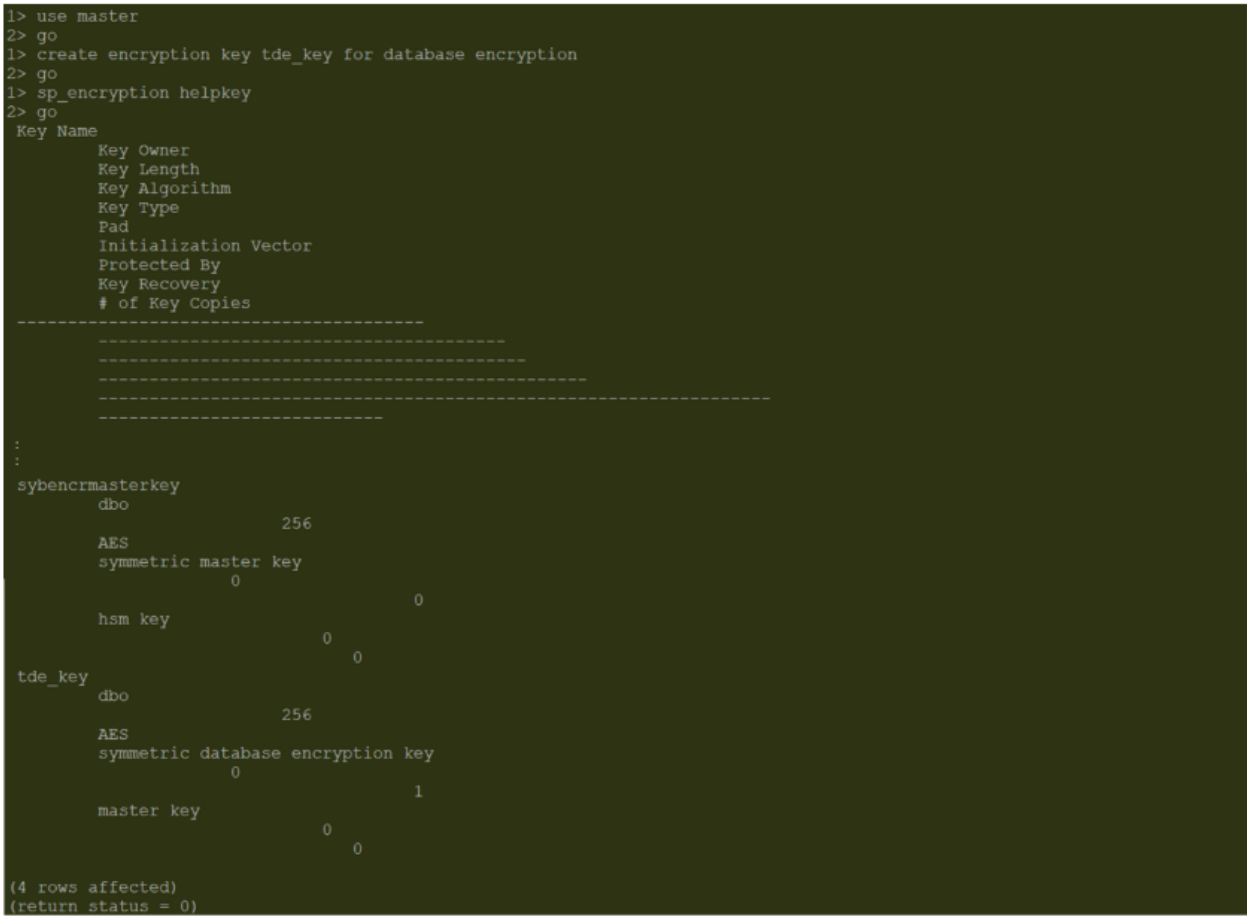

Figure 21 : DEK Details

6. Encrypt the newly created unencrypted database using the alter database command:

```
1>alter database tde_test encrypt with tde_key
2>go
```

```
1> alter database tde_test encrypt with tde_key
2> go
Initiated encryption tasks on database tde_test in background.
1> select name, status
2> from master..sysdatabases
3> where name = 'tde_test'
4> go
 name                                                             status
 ---------------------------------------------------------------- ------
 tde_test                                                              0

(1 row affected)
```

Figure 22 : Database Encryption

6.1.2 Database Encryption Verification

1. Restart the ASE server and verify encrypted database access:

Shutdown the ASE server then restart the ASE server using `startserver`.

```
1>shutdown
2>go
```

```
$/opt/sap/ASE-16_1/bin/startserver -f RUN_SAP01
```

Verify the ASE server log (`/opt/sap/ASE-16_1/install/SAP01.log`) and confirm that the encrypted database startup is having no issues.

```
00:0006:00000:00002:2026/03/24 10:45:55.47 server  The transaction log in the database 'pubs3' will use I/O size of 4 Kb.
00:0006:00000:00002:2026/03/24 10:45:55.47 server  Database 'pubs3' is now online.
00:0006:00000:00002:2026/03/24 10:45:55.63 server  Recovering database 'tde_test' (dbid 6).
00:0006:00000:00002:2026/03/24 10:45:55.63 server  Started estimating recovery log boundaries for database 'tde_test'.
00:0006:00000:00002:2026/03/24 10:45:55.64 server  Database 'tde_test', checkpoint=(980, 21), first=(980, 21), last=(980, 21).
00:0006:00000:00002:2026/03/24 10:45:55.64 server  Completed estimating recovery log boundaries for database 'tde_test'.
00:0006:00000:00002:2026/03/24 10:45:55.64 server  Started ANALYSIS pass for database 'tde_test'.
00:0006:00000:00002:2026/03/24 10:45:55.64 server  Completed ANALYSIS pass for database 'tde_test'.
00:0006:00000:00002:2026/03/24 10:45:55.64 server  Log contains all committed transactions until 2026/03/24 10:03:57.44 for database t
de_test.
00:0006:00000:00002:2026/03/24 10:45:55.64 server  Started REDO pass for database 'tde_test'. The total number of log records to proce
ss is 1.
00:0006:00000:00002:2026/03/24 10:45:55.64 server  Redo pass for database 'tde_test': 1 records done (100%); 0 records left.
00:0006:00000:00002:2026/03/24 10:45:55.64 server  Completed REDO pass for database 'tde_test'.
00:0006:00000:00002:2026/03/24 10:45:55.64 server  Timestamp for database 'tde_test' is (0x0000, 0x000018be).
00:0006:00000:00002:2026/03/24 10:45:55.64 server  Recovery of database 'tde_test' will undo incomplete nested top actions.
00:0006:00000:00002:2026/03/24 10:45:55.64 server  Started recovery checkpoint for database 'tde_test'.
00:0006:00000:00002:2026/03/24 10:45:55.64 server  Completed recovery checkpoint for database 'tde_test'.
00:0006:00000:00002:2026/03/24 10:45:55.64 server  Started filling free space info for database 'tde_test'.
00:0006:00000:00002:2026/03/24 10:45:55.64 server  Completed filling free space info for database 'tde_test'.
00:0006:00000:00002:2026/03/24 10:45:55.65 server  Started cleaning up the default data cache for database 'tde_test'.
00:0006:00000:00002:2026/03/24 10:45:55.65 server  Completed cleaning up the default data cache for database 'tde_test'.
00:0006:00000:00002:2026/03/24 10:45:55.65 server  Checking external objects.
00:0006:00000:00002:2026/03/24 10:45:55.65 server  The transaction log in the database 'tde_test' will use I/O size of 4 Kb.
00:0006:00000:00002:2026/03/24 10:45:55.65 server  Database 'tde_test' is now online.
00:0006:00000:00002:2026/03/24 10:45:55.66 server  Recovering database 'sybmgmtdb' (dbid 31515).
00:0006:00000:00002:2026/03/24 10:45:55.66 server  Started estimating recovery log boundaries for database 'sybmgmtdb'.
```

Figure 23 : ASE Server Log -ASE Server Started With No Issues

2. Verify the encrypted database access:

```
1>use tde_test
2>go
1>select * from emp_details
2>go
```

```
[sap@master-node ~]$ isql -SSAP01 -Usa
Password:
1> use tde_test
2> go
1> select * from emp_details
2> go
 emp_id
        emp_name

        designation
 -----------
        ------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------
        ------------------------------------------------------------------------------------------------------------------------
          1
        Alex

        Engineer
          2
        John

        Finance
          3
        Mark

        Manager

(3 rows affected)
```

Figure 24 : Encrypted Database

3. Shutdown the HSM, restart the ASE server and verify the ASE server logs.

ASE server logs (`/opt/sap/ASE-16_1/install/SAP01.log`) will have the PKCS11 error for the encrypted database startup.

```
00:0006:00000:00002:2026/03/24 09:57:09.70 server  Completed filling free space info for database 'pubs3'.
00:0006:00000:00002:2026/03/24 09:57:09.71 server  Started cleaning up the default data cache for database 'pubs3'.
00:0006:00000:00002:2026/03/24 09:57:09.71 server  Completed cleaning up the default data cache for database 'pubs3'.
00:0006:00000:00002:2026/03/24 09:57:09.71 server  Checking external objects.
00:0006:00000:00002:2026/03/24 09:57:09.71 server  The transaction log in the database 'pubs3' will use I/O size of 4 Kb.
00:0006:00000:00002:2026/03/24 09:57:09.71 server  Database 'pubs3' is now online.
00:0005:00000:00000:2026/03/24 09:57:09.72 kernel  Call to Cryptoki API 'C_OpenSession' failed with error code 5 (0x5) CKR
_GENERAL_ERROR.
00:0006:00000:00002:2026/03/24 09:57:09.72 server  Error: 15432, Severity: 16, State: 15
00:0006:00000:00002:2026/03/24 09:57:09.72 server  A validation check failed when Adaptive Server decrypted an encryption
key. This error may indicate an incorrect password.
00:0006:00000:00002:2026/03/24 09:57:09.72 server  Error: 975, Severity: 17, State: 1
00:0006:00000:00002:2026/03/24 09:57:09.72 server  The database 'tde_test' is encrypted.
00:0006:00000:00002:2026/03/24 09:57:09.72 server  Error: 3419, Severity: 17, State: 1
00:0006:00000:00002:2026/03/24 09:57:09.72 server  Unable to proceed with the recovery of dbid <6> because of previous err
ors.
00:0006:00000:00002:2026/03/24 09:57:09.72 server  Continuing recovery with the next database.
00:0006:00000:00002:2026/03/24 09:57:09.72 server  Recovering database 'sybmgmtdb' (dbid 31515).
00:0006:00000:00002:2026/03/24 09:57:09.72 server  Started estimating recovery log boundaries for database 'sybmgmtdb'.
00:0006:00000:00002:2026/03/24 09:57:09.72 server  Database 'sybmgmtdb', checkpoint=(19280, 30), first=(19280, 30), last=(
19280, 30).
00:0006:00000:00002:2026/03/24 09:57:09.72 server  Completed estimating recovery log boundaries for database 'sybmgmtdb'.
00:0006:00000:00002:2026/03/24 09:57:09.73 server  Started ANALYSIS pass for database 'sybmgmtdb'.
00:0006:00000:00002:2026/03/24 09:57:09.73 server  Completed ANALYSIS pass for database 'sybmgmtdb'.
00:0006:00000:00002:2026/03/24 09:57:09.73 server  Log contains all committed transactions until 2026/03/24 09:55:31.23 fo
r database sybmgmtdb.
00:0006:00000:00002:2026/03/24 09:57:09.73 server  Started REDO pass for database 'sybmgmtdb'. The total number of log rec
ords to process is 1.
```

Figure 25 : ASE Server Log- Encrypted Database Startup Failure

4. Verify the encrypted database.

```
[sap@master-node ~]$ isql -SSAP01 -Usa
Password:
1> use tde_test
2> go
Msg 15432, Level 16, State 15:
Server 'SAP01', Line 1:
A validation check failed when Adaptive Server decrypted an encryption key. This
error may indicate an incorrect password.
Msg 975, Level 17, State 1:
Server 'SAP01', Line 1:
The database 'tde_test' is encrypted.
```

Figure 26 : Encrypted Database Access Failure

5. Connect the HSM and restart the ASE server then verify the encrypted database access.

```
[sap@master-node ~]$ isql -SSAP01 -Usa
Password:
1> use tde_test
2> go
1> select * from emp_details
2> go
 emp_id
        emp_name

        designation
 -----------
        ------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------
        ----------------------------------------------------------------------------------------------------
          1
        Alex

        Engineer
          2
        John

        Finance
          3
        Mark

        Manager

(3 rows affected)
```

Figure 27 : Encrypted Database Access After HSM Connection

7 Optional Features

7.1 Key Replacement Procedure

SAP Sybase ASE supports internal key rotation only and does not natively support external HSM key rotation. A limitation was identified whereby SAP Sybase ASE does not allow generation of a new HSM key while an existing HSM key is already configured and actively referenced by the ASE server. As external HSM key rotation is not supported by SAP Sybase ASE, verification was therefore performed using the alternative approach. The database will initially encrypted using a Data Encryption Key (DEK) protected by a master key associated with the current HSM key, the database will be decrypted using the existing DEK. Then DEK, master key and HSM key will be removed from the ASE server respectively. Then a new HSM key, master key and DEK will be created respectively and the database will be re-encrypted using the new DEK protected by the new master key associated with the newly created HSM key.

1. Decrypt the encrypted database:

ASE reads encrypted pages and decrypts them using the DEK then writes them back in plaintext.

```
1>use master
2>go
1>alter database tde_test decrypt
2>go
```

```
1> use master
2> go
1> SELECT name FROM sysdatabases
2> go
 name
 ------------------------------------------------------------
 master
 model
 pubs2
 pubs3
 sybmgmtdb
 sybsystemdb
 sybsystemprocs
 tde_test
 tempdb

(9 rows affected)
1> select name, status
2> from master..sysdatabases
3> where name = 'tde_test'
4> go
 name                                                         status
 ------------------------------------------------------------ ------
 tde_test                                                          0

(1 row affected)
1> alter database tde_test decrypt
2> go
Initiated decryption tasks on database tde test in background.
```

Figure 28 : Database Decrypted

2. Drop the database encryption key (DEK):

The `drop encryption key` command is used to drop the encryption key. It deletes the database encryption key from the `sysencryptkeys` table in the `master` database.

```
1>use master
2>go
1>drop encryption key tde_key
2>go
1>sp_encryption helpkey
2>go
```

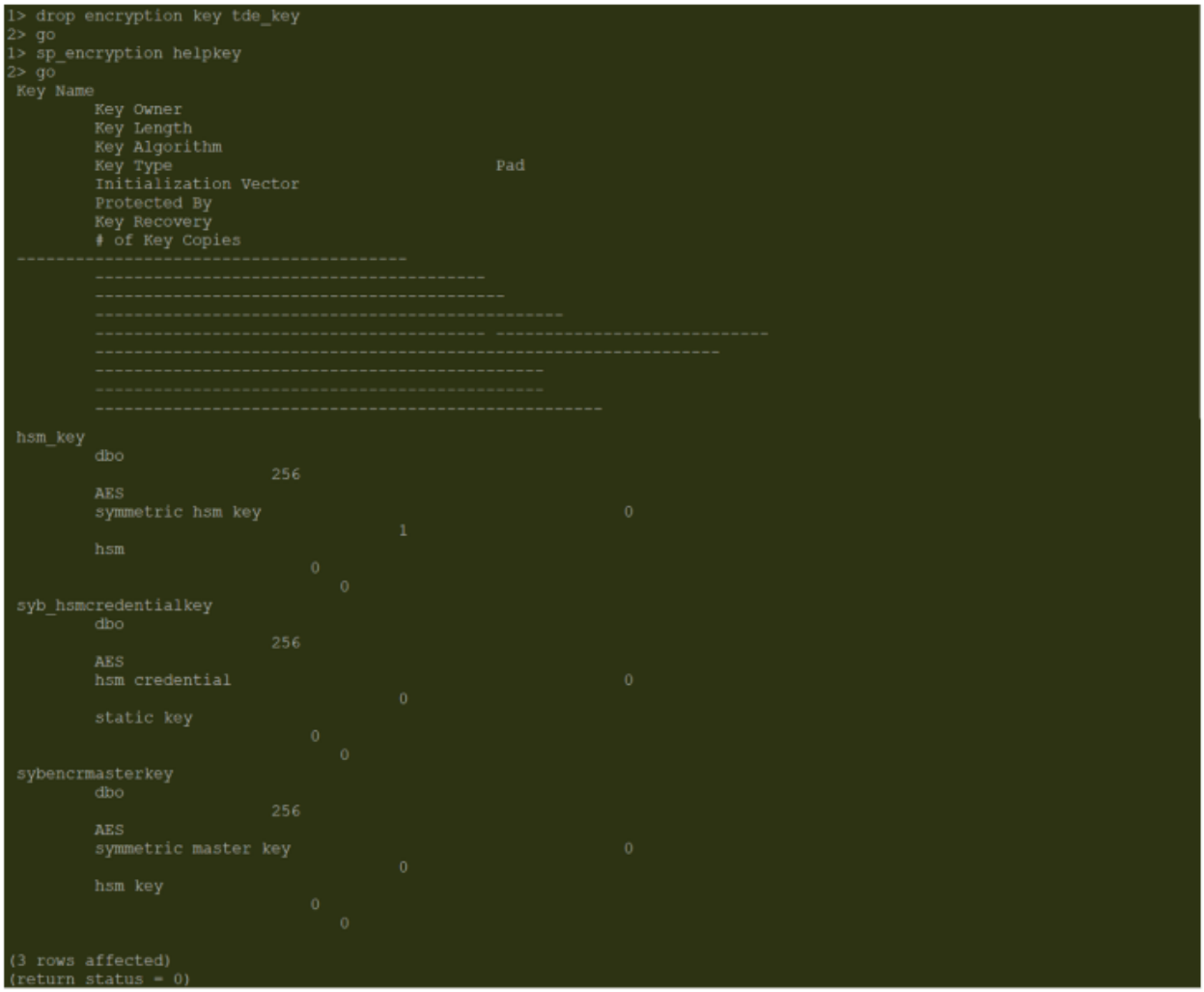

Figure 29 : DEK Dropped

3. Drop the master key:

The `drop encryption key` command is used to drop the master key.

```
1>use master
2>go
1>drop encryption key master
2>go
1>sp_encryption helpkey
2>go
```

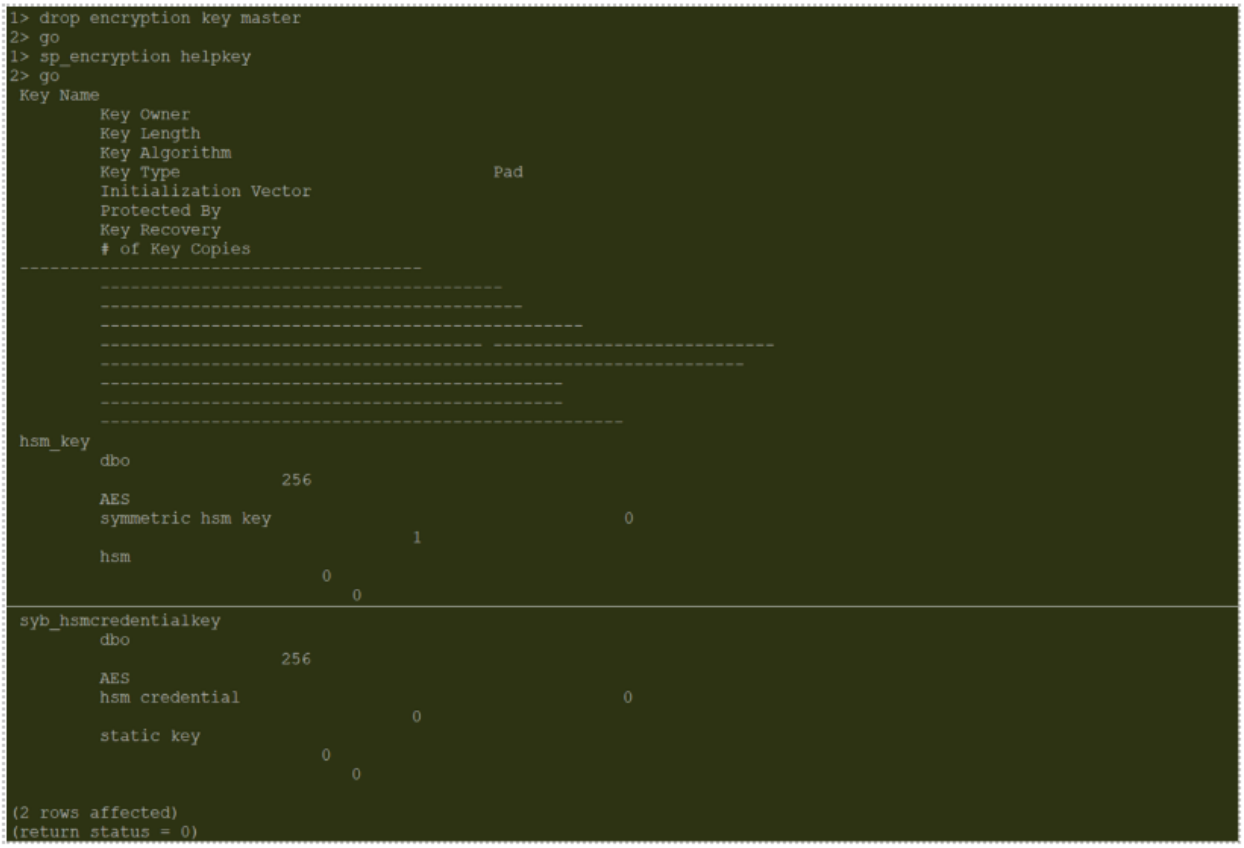

Figure 30 : ASE Master Key Dropped

4. Drop the HSM key:

The HSM key 'hsm_key' will be dropped using `drop encryption key` command. The key will be removed from the HSM as well if we are using the `cascade` argument.

```
1>drop encryption key hsm_key cascade
2>go
1>sp_encryption helpkey
```

```
2>go
```

```
1> drop encryption key hsm_key cascade
2> go
1> sp_encryption helpkey
2> go
 Key Name
         Key Owner
         Key Length
         Key Algorithm
         Key Type                                Pad
         Initialization Vector
         Protected By
         Key Recovery
         # of Key Copies
         UUID
 ---------------------------------------
         ---------------------------------------
         -------------------------------------------
         -----------------------------------------------
         ---------------------------------------- ----------------------------
         ------------------------------------------------------------------
         -----------------------------------------------
         -------------------------------------------------
         -----------------------------------------------------
         ----------------------------
 syb_hsmcredentialkey
         dbo
                          256
         AES
         hsm credential                                      0
                                  0
         static key
                                 0
                                    0
         NULL

(1 row affected)
(return status = 0)
```

Figure 31 : HSM Key Dropped

5. Create a new HSM key:

A new HSM key 'hsm_new_key' is created using `create encryption key` command.

```
1>create encryption key hsm_new_key on external keystore
2>with keylength 256 init_vector random
3>go
1>sp_encryption helpkey
2>go
```

```
1> create encryption key hsm_new_key on external keystore
2> with keylength 256 init_vector random
3> go
1> sp_encryption helpkey
2> go
 Key Name
        Key Owner
        Key Length
        Key Algorithm
        Key Type                                Pad
        Initialization Vector
        Protected By
        Key Recovery
        # of Key Copies
 ----------------------------------------
        ------------------------------------
        ----------------------------------------
        ------------------------------------------------
        ---------------------------------------- ----------------------------
        ------------------------------------------------------------------
        ----------------------------------------------
        ----------------------------------------------
        --------------------------------------------------

 hsm_new_key
        dbo
                            256
        AES
        symmetric hsm key                                       0
                                        1
        hsm
                                    0
                                       0
 syb_hsmcredentialkey
        dbo
                            256
        AES
        hsm credential                                          0
                                        0
        static key
                                    0
                                       0

(2 rows affected)
(return status = 0)
```

Figure 32 : New HSM key 'hsm_new_key' Got Created

6. Verify the new HSM key in ASE Server:

```
1>sp_encryption helpkey,hsm_new_key
2>go
```

```
1> sp_encryption helpkey,hsm_new_key
2> go
 Key Name               Key Owner
        Key Length
        Key Algorithm
        Key Type                                Pad
        Initialization Vector
        Protected By
        Key Recovery
        # of Key Copies
        UUID
 ---------------------- ---------------------------------------
        ----------------------------------------
        ------------------------------------------------
        ---------------------------------------- --------------------------
        ------------------------------------------------------------------
        ------------------------------------------------
        ------------------------------------------------
        ----------------------------------------------------
        ------------------------------------------------------------------
 hsm_new_key            dbo
                           256
        AES
        symmetric hsm key                                       0
                                        1
        hsm
                                 0
                                   0
        029856a2be703b44b5b3b9db1c1d8292

(1 row affected)
(return status = 0)
```

Figure 33 : New HSM Key Available in ASE Server

7. Verify the new HSM key in HSM:

```
$./p11tool2 LoginUser=<Crypto User PIN> ListObjects
```

```
[admin@master-node Administration]$ ./p11tool2 LoginUser=87654321 ListObjects


CKO_SECRET_KEY:

+ 1.1
  CKA_KEY_TYPE                  = CKK_AES
  CKA_UNIQUE_ID                 = 4272B9F1-2B25-46CA-9369-643C8584847E
  CKA_SENSITIVE                 = CK_TRUE
  CKA_EXTRACTABLE               = CK_FALSE
  CKA_LABEL                     = SAP ASE instance key
  CKA_ID                        =
                0x30323938 35366132 62653730 33623434 |029856a2be703b44|
                  62356233 62396462 31633164 38323932 |b5b3b9db1c1d8292|

[admin@master-node Administration]$ cd
```

Figure 34 : New HSM Key Update in HSM

8. Create a new Master key:

```
1>create encryption key master with external key
2>go
```

```
1>sp_encryption helpkey
2>go
```

```
1> create encryption key master with external key
2> go
1> sp_encryption helpkey
2> go
 Key Name
        Key Owner
        Key Length
        Key Algorithm
        Key Type                                Pad
        Initialization Vector
        Protected By
        Key Recovery
        # of Key Copies
 ------------------------------------
        ----------------------------------------
        ----------------------------------------
        ------------------------------------------------
        ---------------------------------------- ----------------------------
        ------------------------------------------------------------------------
        ------------------------------------------------
        ----------------------------------------------
        --------------------------------------------------

 hsm_new_key
        dbo
                        256
        AES
        symmetric hsm key                                       0
                                    1
        hsm
                                0
                                    0
 syb_hsmcredentialkey
        dbo
                        256
        AES
        hsm credential                                          0
                                    0
        static key
                                0
                                    0
 sybencrmasterkey
        dbo
                        256
        AES
        symmetric master key                                    0
                                    0
        hsm key
                                0
                                    0

(3 rows affected)
(return status = 0)
```

Figure 35 : New Master Key Created

9. Create a Database Encryption Key (DEK) to encrypt the database:

```
1>create encryption key tde_key for database encryption
2>go
1>sp_encryption helpkey
2>go
```

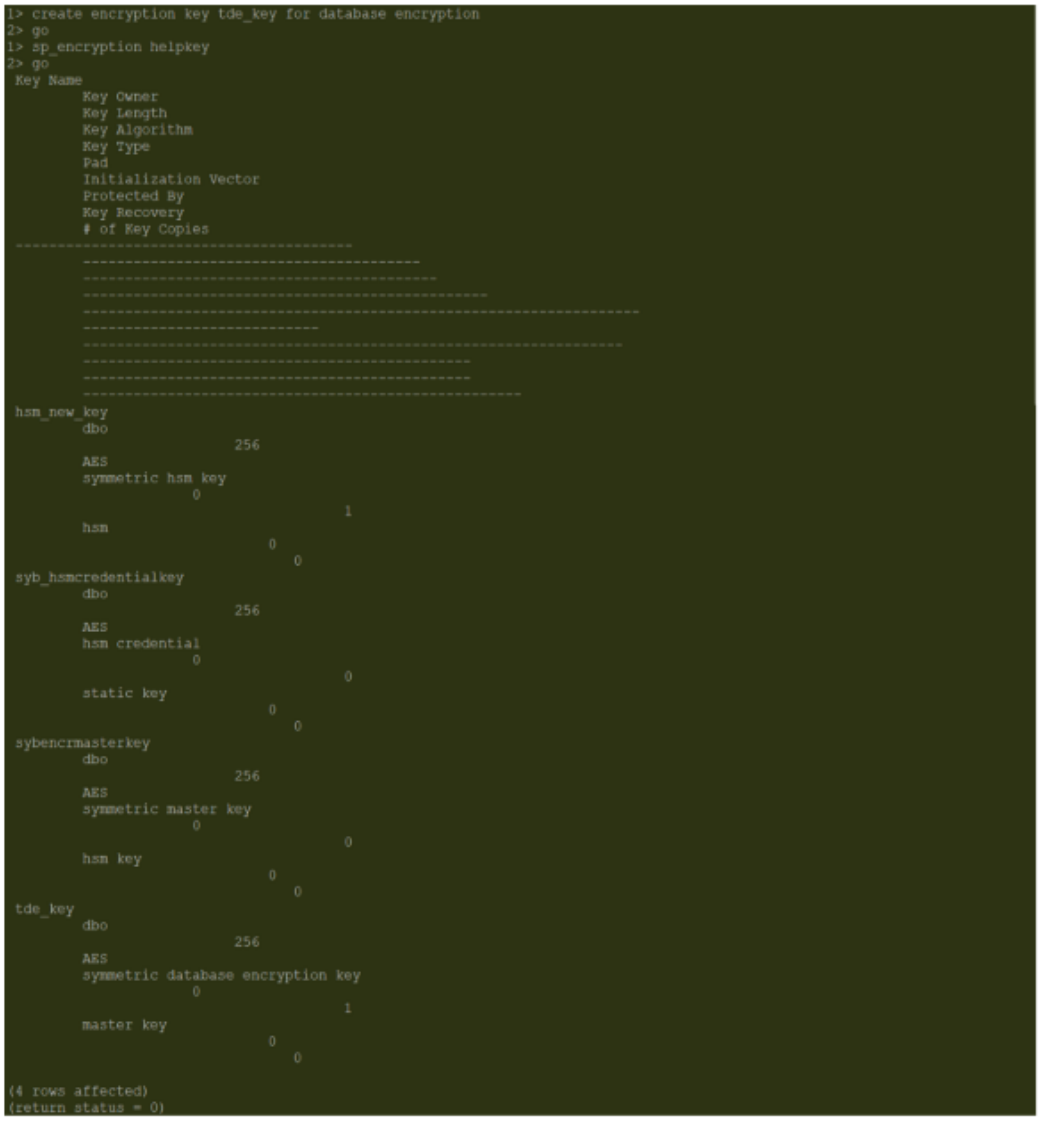

Figure 36 : New DEK Created

10. Turn on the encryption for the database and verify the status of encrypted database:

Status of database encryption can be identified by running the `dbencryption_status()` built-in function. The status will be 0 if the database is not encrypted and status will be 1 if the database is encrypted.

```
1>alter database tde_test encrypt with tde_key
2>go
1>select dbencryption_status('status',db_id('tde_test')) as tde_status
```

```
2>go
```

```
1> alter database tde_test encrypt with tde_key
2> go
Initiated encryption tasks on database tde_test in background.
1> select dbencryption_status('status', db_id('tde_test')) as tde_status
2> go
 tde_status
 ----------
          1

(1 row affected)
```

Figure 37 : Database Encrypted

11. Verify encrypted database 'tde_test' access after ASE server restart:

Shutdown the ASE server then restart the ASE server using `startserver`.

```
1>shutdown
2>go
```

```
$cd ASE-16_1/install
$/opt/sap/ASE16_1/bin/startserver -f RUN_SAP01
```

```
1> shutdown
2> go
Server SHUTDOWN by request.
ASE is terminating this process.
CT-LIBRARY error:
        ct_results(): network packet layer: internal net library error: Net-Library operation terminated due to disconne
[sap@master-node ~]$ cd ASE-16_1/install/
[sap@master-node install]$ /opt/sap/ASE-16_1/bin/startserver -f RUN_SAP01
[sap@master-node install]$ 00:0000:00000:00000:2026/03/24 10:45:52.11 kernel  SySAM: Using licenses from: /usr/local/fle
m/licenses/license.dat
00:0000:00000:00000:2026/03/24 10:45:52.12 kernel  SySAM: Checked out graced license for 1 ASE_CORE (2025.0903) will exp
e Thu 16 Apr 2026 03:45:56 AM PDT.
00:0000:00000:00000:2026/03/24 10:45:52.12 kernel  SySAM: Failed to obtain license(s) for ASE_CORE feature from license
le(s) or server(s).
00:0000:00000:00000:2026/03/24 10:45:52.12 kernel  SySAM: Cannot find license file.
00:0000:00000:00000:2026/03/24 10:45:52.12 kernel  SySAM:  The license files (or license server system network addresses
attempted are listed below.
00:0000:00000:00000:2026/03/24 10:45:52.12 kernel  SySAM: License feature name:  ASE_CORE
00:0000:00000:00000:2026/03/24 10:45:52.12 kernel  SySAM: License filename:      /opt/sap/SYSAM-2_0/licenses
00:0000:00000:00000:2026/03/24 10:45:52.12 kernel  SySAM: License search path:   /opt/sap/SYSAM-2_0/licenses
```

Figure 38 : ASE Server Restarted

12. Verify encrypted database access after ASE server restart.

```
1>use tde_test
2>go
1>select * from emp_details
2>go
```

```
[sap@master-node ~]$ isql -SSAP01 -Usa
Password:
1> use tde_test
2> go
1> select * from emp_details
2> go
 emp_id
        emp_name

        designation
 -----------
        ------------------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------
        ----------------------------------------------------------------------------------------------------
          1
        Alex

        Engineer
          2
        John

        Finance
          3
        Mark

        Manager

(3 rows affected)
```

Figure 39 : Encrypted Database Access

8 Troubleshooting

8.1 Log locations and interpretation

PKCS11 Log File

Log File Name: `cs_pkcs11_R3.log`

Location: Defined by the LogPath parameter in the PKCS#11 configuration file. Example: `\tmp` for Linux.

Details: This log captures detailed information about PKCS#11 operations, including initialization, cryptographic actions, and error messages. The verbosity is controlled by the Logging Loglevel setting.

To simplify your testing process, it's recommended that you enable the PKCS#11 log file by adjusting the logging settings.

Specifically: Set the LogPath to a writable directory (not a specific file). Set the Logging Loglevel to 1 for basic logging. Increase it to 4 for more detailed output during testing. This will generate a log file named `cs_pkcs11_R3.log` within the specified LogPath directory. Reviewing this log can help with troubleshooting if you encounter issues.

Once testing is complete, it's advisable to reduce Logging Loglevel to limit output to only critical or important messages

Sybase ASE Log File

Log File Name: `<servername>.log` (eg: `SAP01.log`)

Location: `$SYBASE/$SYBASE_ASE/install` (eg: `/opt/sap/ASE-16_1/install/`)

Details: This log captures low-level interaction details between SAP ASE and the Utimaco PKCS#11 library, providing visibility into the initialization, execution, and failure of cryptographic operations offloaded to the HSM.

9 Contact and Support Information

You can reach us from Monday to Friday, 09.00 a.m. to 05.00 p.m., Central European Time (CET).

Utimaco IS GmbH
Krefelder Straße 220
52070 Aachen
Germany

RMA Query

If you need to send the device back to Utimaco IS GmbH, please open a new RMA case (Return Merchandise Authorization). We request that you use the following web address. RMA cases cannot be opened by email or phone.

https://support.hsm.utimaco.com/support/rma/new

Other Support Queries

- Mail (preferred contact method)
  support@utimaco.com
  Attach the diagnostic information to your email.
- Web portal
  https://support.hsm.utimaco.com/support/cases/new/
  The diagnostic information will be requested in our response if necessary.
- By phone
  AMERICAS +1-844-UTIMACO (+1 844-884-6226)
  EMEA +49 800-627-3081
  APAC +81 800-919-1301
  The diagnostic information will be requested in our response if necessary.

10 Appendices

10.1 References

Title	Description	Document/Link
SAP ASE– Installation Guide for Linux	Provides step-by-step instructions for installing and configuring SAP Adaptive Server Enterprise on supported Linux platforms, including prerequisites, installation methods, and post-installation tasks	https://help.sap.com/docs/SAP_ASE/23c3bb4a29be443ea887fa10871a30f8/a6612e5fbc2b10149d8a80b52f34dc5a.html
SAP ASE – Database Encryption Guide	Describes how to configure and manage full database and column-level encryption in SAP ASE to protect data at rest using encryption keys and key management features.	https://help.sap.com/docs/SAP_ASE/833788dd3e9c413799014a0fd002d0b2/a6648bc9bc2b1014b48885ebe86c6f54.html

Title	Description	Document/Link
SAP ASE – Database Encryption Administration Guide	Provides detailed administrative procedures for implementing, managing, and securing database and column encryption in SAP ASE, including encryption keys and master key management.	https://help.sap.com/doc/a613310dbc2b1014ade0f78b6ecb68ec/16.0.3.6/en-US/SAP_ASE_Database_Encryption_en.pdf

Table 6: References

10.2 Command Summary

Command	Purpose
`./p11tool2 slot=<slot_no> Label=<token_label> Login=ADMIN,ADMIN.key InitToken=<SO_PIN>`	Initialize PKCS#11 token and create Security Officer (SO) credentials
`./p11tool2 slot=<slot_no> LoginSO=<SO_PIN> InitPin=<CryptoUser_PIN>`	Initialize Crypto User PIN for the PKCS#11 slot
`./p11tool2 LoginUser=<CryptoUser_PIN> ListObjects`	List cryptographic objects stored in the HSM
`isql -S<SAP Server Name> -Usa`	Connect to SAP Sybase ASE using ISQL utility
`SELECT @@VERSION`	Display SAP Sybase ASE server version

Command	Purpose
`SELECT @@SERVERNAME`	Display SAP Sybase ASE server name
`shutdown`	Gracefully shut down the SAP Sybase ASE server
`$SYBASE/$SYBASE_ASE/bin/startserver -f RUN_SAP01`	Start the SAP Sybase ASE server name, eg:SAP01
`sp_configure 'enable encrypted columns', 1`	Enable database and column-level encryption features
`sp_configure 'external keystore', 0, 'hsm'`	Configure HSM as external keystore in ASE
`sp_encryption 'hsm_credential' , 'lib=<pkcs11 library name>; pin=<crypto user pin>; slot=<slot number>'`	Store HSM PKCS#11 library, slot and credentials in ASE
`create encryption key master with keylength 256 init_vector random`	Create HSM-backed ASE master encryption key
`sp_encryption helpkey`	Display all encryption keys configured in ASE
`create database <database_name>`	Create database for Transparent Data Encryption testing
`create encryption key <DEK_name> for database encryption`	Create Database Encryption Key (DEK)

Command	Purpose
`alter database <database_name> encrypt with <DEK_name>`	Encrypt database using the DEK
`alter database <database_name> decrypt`	Decrypt the encrypted database
`select dbencryption_status('status', db_id('<database name>'))`	Check encryption status of the database
`drop encryption key <key_name>`	Drop the encryption key
`create encryption key <hsm key name> on external keystore with keylength 256 init_vector random`	Create a new encryption key directly on HSM
`select * from <table name>`	Verify data access before and after encryption

Table 7: Command Summary