

Using Fortanix Data Security Manager with Oracle TDE

Fortanix Data Security Manager + Oracle Transparent Data Encryption (TDE)

Market need

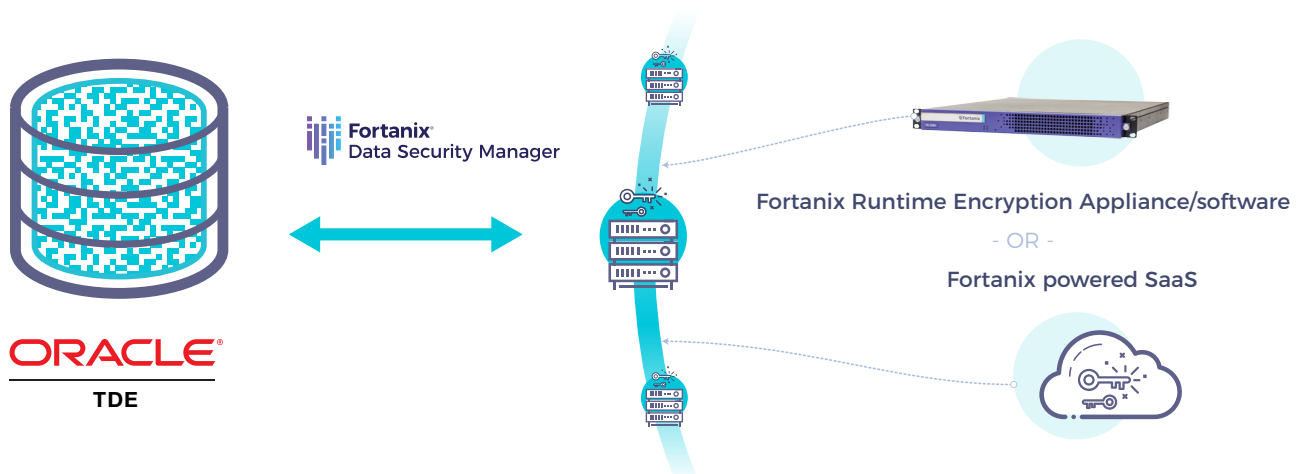


Fortanix Data Security Manager, is a unified HSM and Key Management solution that easily integrates with Oracle Transparent Data Encryption (TDE) and MySQL Data-at-Rest Encryption. The joint Oracle and Fortanix Data Security Manager solution offers scalable data protection and compliance for data center and cloud environments.

Solution Overview

This article describes how to integrate Fortanix Data Security Manager to be used with Oracle TDE. Transparent Data Encryption (TDE) in Oracle Databases allows you to protect sensitive data in tables by encrypting them when they are stored on media. The data is transparently decrypted for authorized users or applications when they access the data. See Introduction to Transparent Data Encryption for more information.

Note: Fortanix Data Security Manager works with Oracle versions 11, 12.1 and 12.2. The following instructions are applicable and have been verified with Oracle Database version 12c release 2 running on CentOS 7.2.



Configuration



Please also refer to the video demo:
<https://www.youtube.com/watch?v=eMpc2yozFTQ>

Before starting, download the Fortanix Data Security Manager PKCS#11 library from [here](#).
The following steps have been verified with the Oracle Database version 12c release 2 running on CentOS 7.2.

- Add the following line to the `sqlnet.ora` file:

```
ENCRYPTION_WALLET_LOCATION = (SOURCE = (METHOD=HSM))
```

- Copy Fortanix Data Security Manager PKCS#11 library to folder `/opt/oracle/extapi/64/hsm/fortanix/0.8.0`. Change the name of the library file to `libpkcs11.so`.
- Set permission and ownership for the folder that contains the Fortanix Data Security Manager PKCS#11 library

```
[oracle@localhost ~]$ sudo chown -R oracle:oinstall /opt/oracle  
[oracle@localhost ~]$ sudo chmod -R 775 /opt/oracle
```

- Add an application to Fortanix Data Security Manager for Oracle TDE. See Getting Started. Copy the API key for the application and write it to a file along with the API endpoint. This file can also be used to specify the location of a file that the PKCS#11 library can write logs to. Note the absolute path of this file, you will need it later. For example, file `dsm_app_api_key` could have:

```
api_endpoint = "https://dsm.fortanix.com"  
api_key = "MWY5YT...T05n"  
[log]  
file = "<log filename>"
```

- Run `sqlplus` and then login with user `sys` with role of `sysdba`

```
[oracle@localhost ~]$ sqlplus  
SQL*Plus: Release 12.2.0.1.0 Production on Fri Dec 1 00:47:19 2017  
Copyright (c) 1982, 2016, Oracle. All rights reserved.  
SQL> connect  
Enter user-name: sys as sysdba  
Enter password:  
Connected.
```

- Switch to root container. Keystore needs to be opened in the root container first. Run the following command in `sqlplus`:

```
SQL> ALTER SESSION SET CONTAINER = CDB$ROOT;
```

- Run the following command to open the hardware key store

```
SQL> ADMINISTER KEY MANAGEMENT SET KEYSTORE OPEN IDENTIFIED BY "file:///home/oracle/dsm_app_api_key"  
CONTAINER = ALL;
```

- After you have opened the hardware keystore, run the following command to set the TDE master encryption key.

```
SQL> ADMINISTER KEY MANAGEMENT SET KEY IDENTIFIED BY "file:///home/oracle/dsm_app_api_key" CONTAINER = ALL;
```

- Connect to sql as a non-sysadmin user to enable encryption on a table column or a tablespace.

```
[oracle@localhost ~]$ sqlplus
SQL\*Plus: Release 12.2.0.1.0 Production on Fri Dec 1 01:08:32 2017
Copyright (c) 1982, 2016, Oracle. All rights reserved.
Enter user-name: user
Enter password:
Connected to:
Oracle Database 12c Enterprise Edition Release 12.2.0.1.0 - 64bit Production
```

- Create a table with an encrypted column.

```
SQL> CREATE TABLE employee (first_name VARCHAR2(128),last_name VARCHAR2(128),empID NUMBER,salary NUMBER(6)
ENCRYPT);
```

- Insert some data into the table

```
SQL> INSERT INTO employee VALUES ('JOHN', 'SMITH',001, 10000);
```

At this point, look at the audit log for the master key in the Fortanix Data Security Manager web UI, and see that the master key was used.

- To list the encrypted columns in your database, run the following command:

```
SQL> select * from dba_encrypted_columns;
```

Fortanix Data Security Manager (DSM)

Secured with Intel® SGX, Fortanix Data Security Manager delivers HSM-grade security with software-defined simplicity. Fortanix Data Security Manager provides flexible consumption options - a hardened appliance, HSM as a service, or software running on commodity x86 servers.

Fortanix Data Security Manager offers central management, tamper-proof logging, rich access control, REST APIs and massive scalability. Organizations use Fortanix Data Security Manager to secure their sensitive cloud and traditional applications, including digital payments, PKI systems, IOT applications, silicon manufacturing, and remote TLS terminations - all while drastically reducing integration complexities and expenses.

[Fortanix Data Security Manager Datasheet](#)
[Fortanix Runtime Encryption Appliance](#)

Oracle Transparent Data Encryption (TDE)

Oracle TDE and MySQL TDE protects your critical data by enabling data-at-rest encryption in the database. It protects the privacy and confidentiality of your information, prevents data breaches and helps meet regulatory compliance requirements. When native encryption is used, the risk is typically transferred to key management systems.