16 Encryption of Sensitive Credential Data in the Data Dictionary

You can encrypt sensitive credential information, such as passwords that are stored in the data dictionary.

- About Encrypting Sensitive Credential Data in the Data Dictionary
  The data dictionary `SYS.LINK$` and `SYS.SCHEDULER$_CREDENTIAL` system tables store sensitive credential data, such as user passwords.
- How the Multitenant Option Affects the Encryption of Sensitive Data
  You can encrypt sensitive data dictionary information from the application root, as well as within individual pluggable databases (PDBs).
- Encrypting Sensitive Credential Data in System Tables
  The `ALTER DATABASE DICTIONARY` statement can encrypt sensitive credential data in the `SYS.LINK$` and `SYS.SCHEDULER$_CREDENTIAL` system tables.
- Rekeying Sensitive Credential Data in the SYS.LINK$ System Table
  You can use the `ALTER DATABASE DICTIONARY` statement to rekey sensitive credential data in the data dictionary `SYS.LINK$` and `SYS.SCHEDULER$_CREDENTIAL` system tables.
- Deleting Sensitive Credential Data in System Tables
  The `ALTER DATABASE DICTIONARY` statement can invalidate existing credentials in `SYS.LINK$` and `SYS.SCHEDULER$_CREDENTIAL` and obfuscate future credential entries to those tables.
- Restoring the Functioning of Database Links After a Lost Keystore
  Database links can be adversely affected if the TDE keystore and its master encryption key is inadvertently lost.
- Data Dictionary Views for Encrypted Data Dictionary Credentials
  Oracle Database provides a set of data dictionary views that provide information about the encryption of sensitive credential data in the data dictionary.

16.1 About Encrypting Sensitive Credential Data in the Data Dictionary

The data dictionary `SYS.LINK$` and `SYS.SCHEDULER$_CREDENTIAL` system tables store sensitive credential data, such as user passwords.

The `SYS.LINK$` table stores information about database links. `SYS.SCHEDULER$_CREDENTIAL` stores information about Oracle Scheduler events. By default, the sensitive credential data stored in these tables is obfuscated.

You can manually encrypt the data that is stored in the `SYS.LINK$` and `SYS.SCHEDULER$_CREDENTIAL` tables by using the `ALTER DATABASE DICTIONARY` statement. Though this feature makes use of Transparent Data Encryption (TDE), you do not need to have an Advanced Security Option license to perform the encryption, but you must have the `SYSKM` administrative privilege. TDE performs the encryption by using the AES256 (Advanced

Encryption Standard) algorithm. The encryption follows the same behavior as other data that is encrypted using TDE.

As a best security practice, Oracle recommends that you encrypt this sensitive credential data. To check the status the data dictionary credentials, you can query the `DICTIONARY_CREDENTIALS_ENCRYPT` data dictionary view.

16.2 How the Multitenant Option Affects the Encryption of Sensitive Data

You can encrypt sensitive data dictionary information from the application root, as well as within individual pluggable databases (PDBs).

When you encrypt, rekey, or decrypt sensitive credential data in the `SYS.LINK$` and `SYS.SCHEDULER$_CREDENTIAL` system tables, you must synchronize the affected PDBs after you complete the process. The instructions for doing so are in the procedures that cover these topics.

16.3 Encrypting Sensitive Credential Data in System Tables

The `ALTER DATABASE DICTIONARY` statement can encrypt sensitive credential data in the `SYS.LINK$` and `SYS.SCHEDULER$_CREDENTIAL` system tables.

The database must have an open keystore and an encryption key before you run the `ALTER DATABASE DICTIONARY` statement with the `ENCRYPT CREDENTIALS` clause to encrypt `SYS.LINK$` and `SYS.SCHEDULER$_CREDENTIAL`. The credential data encryption process de-obfuscates the obfuscated passwords and then encrypts them. The encryption applies to any future password changes that users may make after you complete this process.

1. Connect to either the application root or to a pluggable database (PDB) as a user who as been granted the `SYSKM` administrative privilege.

   For example:

   ```
   CONNECT hr_admin@pdb_name AS SYSKM
   Enter password: password
   ```

   To find the available PDBs in a CDB, log in to the CDB root container and then query the `PDB_NAME` column of the `DBA_PDBS` data dictionary view. To check the current container, run the `show con_name` command.

2. If necessary, create and open a keystore and then set an encryption key.

   You can query the `V$ENCRYPTION_WALLET` dynamic view to find the status of a keystore.

   Use the `ADMINISTER KEY MANAGEMENT` statement to perform these three tasks. For example:

   ```
   ADMINISTER KEY MANAGEMENT CREATE KEYSTORE '/etc/ORACLE/WALLETS/orcl' IDENTIFIED BY
   password;
   ADMINISTER KEY MANAGEMENT SET KEYSTORE OPEN IDENTIFIED BY "password";
   ADMINISTER KEY MANAGEMENT SET ENCRYPTION KEY IDENTIFIED BY "password" WITH BACKUP;
   ```

   Include the `CONTAINER = ALL` clause if you are in the application root. This applies the keystore operation for PDBs that are in united mode. For PDBs that are in isolated mode, run the statement from within the PDB.

3. Run the `ALTER DATABASE DICTIONARY` statement to encrypt the data.

For example:

```
ALTER DATABASE DICTIONARY ENCRYPT CREDENTIALS;
```

In an application root, to apply the encryption to the associated PDBs, include the `CONTAINER = ALL` clause.

```
ALTER DATABASE DICTIONARY ENCRYPT CREDENTIALS CONTAINER = ALL;
```

4. If you performed the encryption from the application root, then synchronize the associated PDBs.

   ```
   ALTER PLUGGABLE DATABASE APPLICATION APP$CDB$SYSTEM SYNC;
   ```

16.4 Rekeying Sensitive Credential Data in the SYS.LINK$ System Table

You can use the `ALTER DATABASE DICTIONARY` statement to rekey sensitive credential data in the data dictionary `SYS.LINK$` and `SYS.SCHEDULER$_CREDENTIAL` system tables.

To rekey this sensitive credential data, you must run the `ALTER DATABASE DICTIONARY` statement with the `REKEY CREDENTIALS` clause. The rekey operation, which uses column encryption, does not affect other TDE master encryption keys.

1. Connect to either the application root or to a pluggable database (PDB) as a user who as been granted the `SYSKM` administrative privilege.

   For example, to connect to a PDB:

   ```
   CONNECT hr_admin@pdb_name AS SYSKM
   Enter password: password
   ```

2. If necessary, create and open a keystore and then set an encryption key.

   You can query the `V$ENCRYPTION_WALLET` dynamic view to find the status of a keystore.

   Use the `ADMINISTER KEY MANAGEMENT` statement to perform these three tasks. For example:

   ```
   ADMINISTER KEY MANAGEMENT CREATE KEYSTORE '/etc/ORACLE/WALLETS/orcl' IDENTIFIED BY
   password;
   ADMINISTER KEY MANAGEMENT SET KEYSTORE OPEN IDENTIFIED BY "password";
   ADMINISTER KEY MANAGEMENT SET ENCRYPTION KEY IDENTIFIED BY "password" WITH BACKUP;
   ```

   Include the `CONTAINER = ALL` clause if you are in the application root.

3. Run the `ALTER DATABASE DICTIONARY` statement to rekey the data.

   For example:

   ```
   ALTER DATABASE DICTIONARY REKEY CREDENTIALS;
   ```

   In an application root, to apply the encryption to the associated PDBs, include the `CONTAINER = ALL` clause.

   ```
   ALTER DATABASE DICTIONARY REKEY CREDENTIALS CONTAINER = ALL;
   ```

4. If you performed the rekey operation from the application root, then synchronize the associated PDBs.

   ```
   ALTER PLUGGABLE DATABASE APPLICATION APP$CDB$SYSTEM SYNC;
   ```

16.5 Deleting Sensitive Credential Data in System Tables

The `ALTER DATABASE DICTIONARY` statement can invalidate existing credentials in `SYS.LINK$` and `SYS.SCHEDULER$_CREDENTIAL` and obfuscate future credential entries to those tables.

To delete this credential data, you must run the `ALTER DATABASE DICTIONARY` statement with the `DELETE CREDENTIALS` clause. This statement is mainly used in cases where you must recover the database link from the loss of a Transparent Data Encryption (TDE) keystore.

1. Connect to either the application root or a pluggable database (PDB) as a user who as been granted the `SYSKM` administrative privilege.

   For example:

   ```
   CONNECT hr_admin@pdb_name AS SYSKM
   Enter password: password
   ```

2. If necessary, create and open a keystore and then set an encryption key.

   You can query the `V$ENCRYPTION_WALLET` dynamic view to find the status of a keystore.

   Use the `ADMINISTER KEY MANAGEMENT` statement to perform these three tasks. For example:

   ```
   ADMINISTER KEY MANAGEMENT CREATE KEYSTORE '/etc/ORACLE/WALLETS/orcl' IDENTIFIED BY
   password;
   ADMINISTER KEY MANAGEMENT SET KEYSTORE OPEN IDENTIFIED BY "password";
   ADMINISTER KEY MANAGEMENT SET ENCRYPTION KEY IDENTIFIED BY "password" WITH BACKUP;
   ```

   Include the `CONTAINER = ALL` clause if you are in the application root.

3. Run the `ALTER DATABASE DICTIONARY` statement to delete the password credential.

   For example:

   ```
   ALTER DATABASE DICTIONARY DELETE CREDENTIALS KEY;
   ```

   In an application root, to delete the `SYS.LINK$` and `SYS.SCHEDULER$_CREDENTIAL` password credentials in the associated PDBs, include the `CONTAINER = ALL` clause.

   ```
   ALTER DATABASE DICTIONARY DELETE CREDENTIALS CONTAINER = ALL;
   ```

4. If you performed the credential deletion from the application root, then synchronize the associated PDBs.

   ```
   ALTER PLUGGABLE DATABASE APPLICATION APP$CDB$SYSTEM SYNC;
   ```

Related Topics

- Restoring the Functioning of Database Links After a Lost Keystore
  Database links can be adversely affected if the TDE keystore and its master encryption key is inadvertently lost.

16.6 Restoring the Functioning of Database Links After a Lost Keystore

Database links can be adversely affected if the TDE keystore and its master encryption key is inadvertently lost.

When a TDE keystore and master encryption key are lost, existing database links that are authenticated with encrypted passwords become unusable.

1. Connect to either the application root or a pluggable database (PDB) as a user who as been granted the `SYSKM` administrative privilege and who has the `ALTER DATABASE LINK` system privilege.

   For example:

   ```
   CONNECT hr_admin@pdb_name AS SYSKM
   Enter password: password
   ```

2. Delete the encrypted credentials from the `SYS.LINK$` system table.

   ```
   ALTER DATABASE DICTIONARY DELETE CREDENTIALS KEY;
   ```

   If you are performing the deletion from the application root, then include the `CONTAINER = ALL` clause.

   ```
   ALTER DATABASE DICTIONARY DELETE CREDENTIALS CONTAINER = ALL;
   ```

3. Create and open a keystore and then set an encryption key.

   For example:

   ```
   ADMINISTER KEY MANAGEMENT CREATE KEYSTORE '/etc/ORACLE/WALLETS/orcl' IDENTIFIED BY
   password;
   ADMINISTER KEY MANAGEMENT SET KEYSTORE OPEN IDENTIFIED BY "password";
   ADMINISTER KEY MANAGEMENT SET ENCRYPTION KEY IDENTIFIED BY "password" WITH BACKUP;
   ```

   Include the `CONTAINER = ALL` clause if you are in the application root.

4. Encrypt the password credentials in `SYS.LINK$` and `SYS.SCHEDULER$_CREDENTIAL`.

   ```
   ALTER DATABASE DICTIONARY ENCRYPT CREDENTIALS;
   ```

   If you are performing the encryption from the application root, then include the `CONTAINER = ALL` clause.

   ```
   ALTER DATABASE DICTIONARY ENCRYPT CREDENTIALS CONTAINER = ALL;
   ```

5. Using the password of the user who is associated with the database link, reset the database link passwords that were affected by the `ALTER DATABASE DICTIONARY DELETE CREDENTIALS KEY` statement.

   For example:

   ```
   ALTER DATABASE LINK database_link_name CONNECT TO user_id IDENTIFIED BY password
   CONTAINER = ALL;
   ```

   To find existing database links and their owners, query the `DBA_DB_LINKS` data dictionary view.

6. If you performed the credential deletion from the application root, then synchronize the associated PDBs.

```
ALTER PLUGGABLE DATABASE APPLICATION APP$CDB$SYSTEM SYNC;
```

16.7 Data Dictionary Views for Encrypted Data Dictionary Credentials

Oracle Database provides a set of data dictionary views that provide information about the encryption of sensitive credential data in the data dictionary.

Table 16-1 lists the data dictionary views.

Table 16-1   Data Dictionary Views for Encrypted Data Dictionary Credentials

View	Description
`ALL_DB_LINKS`	Describes database links that are accessible to the current user. A value of `YES` in the `VALID` column indicates that the database link is usable.
`DBA_DB_LINKS`	Describes describes all database links in the database. A value of `YES` in the `VALID` column indicates that the database link is usable. (This view is available to administrative users only, such as `SYS` or users who have been granted the `DBA` role.)
`DICTIONARY_CREDENTIALS_ENCRYPT`	Describes the status of dictionary credentials. The `ENFORCEMENT` column lists `ENABLED` if the credentials are encrypted and `DISABLED` if the credentials are not encrypted.
`USER_DB_LINKS`	Describes the database links that are owned by the current user. A value of `YES` in the `VALID` column indicates that the database link is usable.

Related Topics

- *Oracle Database Reference*