Chiffrement symétrique des données sous Oracle

Description générale

Le chiffrement symétrique repose sur l'utilisation d'une clé unique pour chiffrer et déchiffrer les données. Avec Oracle, le package DBMS_CRYPTO permet d'utiliser plusieurs algorithmes de chiffrement comme AES, DES et 3DES

Module de chiffrement DBMS_CRYPTO

Oracle fournit le package DBMS_CRYPTO permettant d'effectuer des opérations de chiffrement et de déchiffrement de données de manière sécurisée.

Lien vers le document officiel :
https://docs.oracle.com/en/database/oracle/oracle-database/19/arpls/DBMS_CRYPTO.html

Droit d'exécution sur ce module

Pour utiliser DBMS_CRYPTO, il est nécessaire de disposer du privilège EXECUTE sur le package : `GRANT EXECUTE ON DBMS_CRYPTO TO utilisateur;`

Type des données chiffrées

Les données à chiffrer peuvent être des chaînes de caractères (VARCHAR2, CLOB) ou des types binaires (BLOB). Il est recommandé d'utiliser **RAW** pour le stockage de données chiffrées.

Chiffrement des données

Prenons un exemple sur une base de donnée existante et dans la table Clients, on ajoute une colonne nomCli_enc pour stocker les noms chiffrés :

```
''ALTER TABLE Clients ADD nomCli_enc RAW(2000);''
```

Ensuite on chiffre les noms des clients existants et les stocke dans la colonne nomCli_enc :

```
''UPDATE Clients
SET nomCli_enc = DBMS_CRYPTO.ENCRYPT(
    src => UTL_RAW.cast_to_raw(nomCli),
    typ => DBMS_CRYPTO.ENCRYPT_AES256 + DBMS_CRYPTO.CHAIN_CBC +
```

```
DBMS_CRYPTO.PAD_PKCS5,
    key => UTL_RAW.cast_to_raw('MaCleSecrete16Bytes'),
    iv  => UTL_RAW.cast_to_raw('InitialVector123')
);
''
```

Déchiffrement des données

Un gestionnaire (GESTK) veut récupérer le nom original :

```
SELECT UTL_RAW.CAST_TO_VARCHAR2(
    DBMS_CRYPTO.DECRYPT(
        nomCli_enc,
        DBMS_CRYPTO.ENCRYPT_AES256 + DBMS_CRYPTO.CHAIN_CBC +
DBMS_CRYPTO.PAD_PKCS5,
        UTL_RAW.cast_to_raw('MaCleSecrete16Bytes'),
        UTL_RAW.cast_to_raw('InitialVector123')
    )
) AS nomCli_dechiffre FROM Clients;
```

** On peut également créée une fonction de déchiffrement pour simplifier les requêtes **

```
CREATE OR REPLACE FUNCTION dechiffrer(p_data RAW) RETURN VARCHAR2 AUTHID
CURRENT_USER AS
    v_decrypted RAW(2000);
BEGIN
    v_decrypted := DBMS_CRYPTO.DECRYPT(
        p_data,
        DBMS_CRYPTO.ENCRYPT_AES256 + DBMS_CRYPTO.CHAIN_CBC +
DBMS_CRYPTO.PAD_PKCS5,
        UTL_RAW.cast_to_raw('MaCleSecrete16Bytes'),
        UTL_RAW.cast_to_raw('InitialVector123')
    );
    RETURN UTL_RAW.CAST_TO_VARCHAR2(v_decrypted);
END;
/
```

Donc un gestionnaire peut peut maintenant exécuter une requête simple :

```
SELECT dechiffrer(nomCli_enc) AS nomCli FROM Clients;
```

From:
https://wiki.siochaptalqper.fr/ - **Wiki SIO Chaptal**

Permanent link:
https://wiki.siochaptalqper.fr/doku.php?id=bloc3:bdd-symetric-ciphering&rev=1742283785

Last update: **2025/03/18 08:43**