Overview of Db2 native encryption

Last Updated: 2024-01-12

Db2® native encryption uses a two-tier approach to data encryption. Data is encrypted with a Data Encryption Key (DEK), which is in turn encrypted with a Master Key (MK). The encrypted DEK is stored with the data while the MK is stored in a keystore external to Db2.

Db2 native encryption ensures that the DEK is never exposed outside of the encrypted database, transaction log, or backup file. There are no interfaces provided to access the DEK in either its clear text or encrypted forms. As the MK is stored in a different location from the encrypted data, the chance of the encrypted DEK being concurrently exposed with the MK used to encrypt it is very unlikely. Since the risk of the DEK being exposed is extremely low, the need to rotate it is negligible. The rotation of the MK, which is used to protect the DEK, can be done efficiently without the need to decrypt and re-encrypt the data

Data Encryption Key (DEK)

Db2 encrypts data with a data encryption key (DEK) before the data is written to disk. The DEK is stored, encrypted by the master key (MK), within the database or backup image. The DEK itself is generated by Db2 as needed, such as when an encrypted database or encrypted database backup is created. A unique DEK exists for each encrypted database and for each encrypted backup.

Master Key (MK)

A master key (MK) is an encryption key that is used to encrypt a data encryption key (DEK). Each encrypted database is associated with one master key at one time. Unless directed otherwise, Db2 generates an MK automatically during these operations:

- Database creation
- Master key rotation
- Restoring into a new database

Master keys are identified by a label that Db2 uses to uniquely identify each master key. By default, Db2 creates a label for every new MK created. You can override this behavior by supplying a specific label for a particular MK. Reasons for creating an MK with a particular label include:

- tracking the MK labels and their corresponding keys for offsite recovery without having the entire keystore available on the backup site
- having an HADR pair that requires synchronized keys
- encrypting a backup for an unencrypted database

Keystore

Master keys are stored in a keystore. A keystore can be a file that is directly accessed by Db2 (local) or a third-party keystore with which Db2 communicates over the network (centralized).

> **Note:** A Db2 instance can be configured for one keystore for native encryption at one time.

Keystores supported by Db2

Db2 native encryption can interact with the following keystores:

– A local keystore file that follows the Public Key Cryptography Standards (PKCS) #12 archive file format for storing cryptography objects

> **Note:** PKCS is an OASIS standard for public key cryptography. The numbers 11 and 12 refer to specific parts of the standard.

– A centralized keystore that is accessed using one of the following methods:
  - Any key manager product that supports Key Management Interoperability Protocol (KMIP) version 1.1 or higher. A key manager is software that you can use to create, update, and secure a keystore.

    > **Note:** KMIP is an OASIS standard for network protocol that is related to key management.

  - One of the following supported Hardware Security Modules (HSM) that use the PKCS #11 API:
    - Gemalto Safenet HSM (formerly Luna) version 6.1 (firmware version 6.23.0) and higher
    - Entrust nShield HSM (formerly nCipher, formerly Thales), security world software version 11.50 and higher

MKs and the keystore

An MK can either be created directly within the keystore or generated by Db2, upon request, and stored within the keystore. One or more MKs can exist and each MK can be referenced by different Db2 databases or backup images.

– **Keystore access by Db2 native encryption**
Whenever Db2 needs access to the Data Encryption Key (DEK), the Master Key (MK) is used to decryp the DEK, which requires the keystore to be opened to access the MK. Depending on the type of keystore being used, the MK is either fetched from the keystore into Db2 for decryption of the DEK, o the DEK is shipped to the keystore for decryption.

Parent topic:

→ Db2 native encryption