Implementing TDE in SQL Server

Transparent Data Encryption (TDE) in SQL Server is a feature that encrypts the entire database at rest, including backups and log files. It is not directly part of SQL Server Audit, but you can implement TDE alongside auditing to ensure both data confidentiality and compliance monitoring. Here's how to implement TDE in SQL Server:

Step-by-Step: Implement Transparent Data Encryption (TDE)

1. Create a Master Key

- Connect to your SQL Server instance using SSMS.
- Run the following command to create a database master key:

```
USE master;
CREATE MASTER KEY ENCRYPTION BY PASSWORD = 'YourStrongPassword';
```

2. Create or Obtain a Certificate

- Create a certificate to protect the database encryption key:

```
CREATE CERTIFICATE MyServerCert WITH SUBJECT = 'My Server Certificate';
```

3. Create a Database Encryption Key

- Switch to the target database and create a database encryption key:

```
USE EmployeeDb;
CREATE DATABASE ENCRYPTION KEY
WITH ALGORITHM = AES_256
ENCRYPTION BY SERVER CERTIFICATE MyServerCert;
```

4. Enable TDE

- Enable TDE for your database:

```
ALTER DATABASE YourDatabase SET ENCRYPTION ON;
```

5. Monitor Encryption Progress

- You can check the encryption status using:

```
SELECT db.name, db.is_encrypted, dm.encryption_state, dm.percent_complete
FROM sys.databases db
LEFT JOIN sys.dm_database_encryption_keys dm ON db.database_id = dm.database_id;
```

Notes on Auditing with TDE

- SQL Server Audit can be used to track access and changes to encrypted databases, but TDE itself does not audit who accesses the data.
- To monitor access, use SQL Server Audit specifications to capture login events, permission changes, and queries against encrypted databases.

Best Practices

- Securely back up the certificate and master key. (We have seen this in backup demo)
- Regularly audit access to encrypted databases.
- Use TDE for compliance with data protection regulations.