Verschlüsselung einer Datenspalte und Transparente Datenverschlüsselung

Kathegorie
SQL Server

Aaron
Priesterroth

Im folgenden Artikel möchten wir (ergänzend zu dem hier beschriebenen **Always Encrypted**-Verfahren) zwei weitere Verschlüsselungsmethoden des SQL Server genauer betrachten: zum einen die Verschlüsselung einer einzelnen Datenspalte, zum anderen die Anwendung der transparenten Datenverschlüsselung.

Verschlüsselung einer einzelnen Datenspalte

Die Verschlüsselung von Datenspalten (engl. Column-level encryption) ermöglicht dem Benutzer die Verschlüsselung einzelner Spalten einer Tabelle, ohne dabei die gesamte Datenbank verschlüsseln zu müssen. Der Vorteil hierbei ist die erhöhte Flexibilität in der Selektion der Attribute, die verschlüsselt werden sollen. Das Resultat ist die Minimierung der Ressourcen die beim Einfügen oder Auslesen von Daten für das Ver- und Entschlüsseln benötigt werden.

Vorteile

- Für jede verschlüsselte Spalte wird ein eigener Schlüssel verwendet. Dies minimiert die Wahrscheinlichkeit von unautorisierten Zugängen.
- Flexibilität in der Verschlüsselung der Daten: eine Anwendung kann steuern, welche Daten wann, wo und von wem ausgelesen werden.
- Transparente Verschlüsselung ist möglich.
- Verschlüsselung von "aktiven" Daten, also nicht nur von Daten, die nicht in Verwendung sind.
- Geringe zusätzliche Latenz.

Nachteile

- Limitiert bzw. verringert die möglichen Abfrage-Optimierungen.
- Erhöhter Verbrauch von System-Ressourcen.
- Erhöhtes Potential von Sicherheitslücken.
- Erhöhte Speichernutzung.

Mainzer Datenfabrik GmbH
Rotterdamer Platz 2, 55120 Mainz
info@madafa.de
madafa.de
+49 6131 3331612

Verwendung

Erstellen eines Datenbank-Hauptschlüssels

Mit dem folgenden T-SQL Befehl kann ein neuer Datenbank-Hauptschlüssel erzeugt werden:

```
USE [master]
GO;

CREATE MASTER KEY ENCRYPTION BY PASSWORD = 'SecurePassword123?';
```

Erstellen eines neuen Zertifikats

Mit dem folgenden T-SQL Befehl kann ein neues Zertifikat erzeugt werden:

```
CREATE CERTIFICATE MyCertificate
  WITH SUBJECT = 'MyCertificate Subject';
```

Erstellen eines neuen Schlüssels

Mit dem folgenden T-SQL Befehl kann ein neuer Schlüssel erzeugt werden:

```
CREATE SYMMETRIC KEY MyKey
  WITH ALGORITHM = AES_256
  ENCRYPTION BY CERTIFICATE MyCertificate;
```

Verschlüsseln einer Datenspalte

Mit dem folgenden T-SQL Befehl kann eine Datenspalte verschlüsselt werden:

```
-- Öffne den Schlüssel
OPEN SYMMETRIC KEY MyKey
  DECRYPTION BY CERTIFICATE MyCertificate;

-- Verschlüsselung der Datenspalte "MyColumn" mit Hilfe der Spatel
"MyColumnEncrypted"
UPDATE dbo.MyTable
SET MyColumnEncrypted = EncryptByKey(Key_GUID('MyKey'),
    MyColumn,
    1,
    HashBytes('SHA1', CONVERT(varbinary, MyID)));
GO
```

Transparente Datenverschlüsselung (TDE)

Die Transparente Datenverschlüsselung (engl. **T**ransparent **D**ata **E**ncryption (**TDE**)) wird für die Echtzeit-Verschlüsselung von Ein- und Ausgehenden Daten und Log-Dateien verwendet. Die Verschlüsselung selbst basiert dabei auf einem Datenbank-Schlüssel, der im Boot-Sektor der Datenbank gespeichert wird. Dies dient der Verfügbarkeit des Schlüssels im Rahmen einer möglichen Wiederherstellung (engl. recovery).

Beim Datenbank-Schlüssel handelt es sich entweder

- um einen symmetrischen Schlüssel, der auf einem in der Datenbank zur Verfügung stehenden Zertifikat basiert, oder
- um einen asymmetrischen Schlüssel, der von einem EKM-Modul (Extensible Key Management) geschützt wird.

Hinweise

- TDE kann nur Daten verschlüsseln, die nicht in ”aktiver“ Verwendung sind.
- Die Verschlüsselung der Datenbanken-Datei passiert auf der Page-Ebene des Betriebssystems. Das bedeutet, dass die Seiten einer verschlüsselten Datenbank verschlüsselt werden **bevor** sie auf die Festplatte geschrieben werden, und entschlüsselt werden **bevor** sie in den Arbeitsspeicher geladen werden.

Verwendung →

Mainzer Datenfabrik GmbH
Rotterdamer Platz 2, 55120 Mainz
info@madafa.de
madafa.de
+49 6131 3331612

Verwendung

Erstellen eines Meister-Schlüssels

Mit dem folgenden T-SQL Befehl kann ein neuer Meister-Schlüssel erzeugt werden:

```
USE [master]
GO;

CREATE MASTER KEY ENCRYPTION BY PASSWORD = 'SecurePassword123?';
```

Erstellen eines neuen Zertifikats

Mit dem folgenden T-SQL Befehl kann ein neues Zertifikat erzeugt werden:

```
CREATE CERTIFICATE MyCertificate
   WITH SUBJECT = 'MyCertificate Subject';
```

Erstellen eines neuen Datenbank-Schlüssels

Mit dem folgenden T-SQL Befehl kann ein neuer Datenbank-Schlüssel erzeugt werden:

```
USE []
GO;

CREATE DATABASE ENCRYPTION KEY
  WITH ALGORITHM = AES_256
  ENCRYPTION BY SERVER CERTIFICATE MyCertificate;
```

Aktivieren der Datenbank-Verschlüsselung

Mit dem folgenden T-SQL Befehl kann die TDE-Verschlüsselung auf einer konfigurierten Datenbank aktiviert werden:

```
ALTER DATABASE [<MyDatabase>] SET ENCRYPTION ON;
```

Fazit

Neben der Verschlüsselung einer einzelnen Datenspalte und der Verwendung von Transparenter Verschlüsselung, gibt es unter SQL Server zusätzlich die Möglichkeit das Feature **Always Encrypted zu verwenden**, das auf einem externen Treiber basiert. Mehr Informationen zur Verwendung von Always Encrypted findest Du hier.

certificate, column encryption, encryption, mssql, schlüssel, sql server, sql server 2019, tde, transparent data encryption, transparente verschlüsselung, verschlüsselung, zertifikat

Mainzer Datenfabrik GmbH
Rotterdamer Platz 2, 55120 Mainz
info@madafa.de
madafa.de
+49 6131 3331612