



Access und SQL Server-Verschlüsselung

AEK 22

Nürnberg/Hannover – Herbst 2019

berndjungbluth.de

Zur Person

- Bernd Jungbluth
- Freiberuflicher Berater und Entwickler
 - Migration Access nach SQL Server
 - Administration SQL Server
 - Entwicklung und Optimierung von SQL Server-Datenbanken
 - Datawarehouse-Systeme
 - SQL Server Reporting Services
 - SQL Server Integration Services
 - SQL Server Sicherheitsanalysen
- Zertifizierter Datenschutzbeauftragter

Agenda

- Verschlüsselung
 - Sinn und Zweck
 - Verschlüsselung mit SQL Server
- Verschlüsseln von Daten
 - T-SQL-Befehle
 - Always Encrypted
- Verschlüsselung auf Ebene der SQL Server-Instanz
 - Datenbanksicherung
 - Netzwerkübertragung
- Zusammenfassung und Fazit

Sinn und Zweck

■ Verschlüsselung

- Eine Wissenschaft für sich: »Kryptologie«
 - „von einem Schlüssel abhängige Umwandlung von ‚Klartext‘ genannten Daten in einen ‚Geheimtext‘“ (Quelle: wikipedia.de)
 - Ohne Schlüssel keine Umwandlung von Geheimtext in Klartext
- Mehrere Methoden zur Verschlüsselung
 - Symmetrische Schlüssel, Asymmetrische Schlüssel und Zertifikate

■ Notwendigkeit

- Zum technischen Schutz von Unternehmensdaten
 - Betriebsgeheimnisse wie Rezepturen, Forschungsergebnisse, Patente etc.
 - Personenbezogene Daten
- Erhöht die Datensicherheit und die IT-Sicherheit

Sinn und Zweck / Verschlüsselungsmethoden

■ Symmetrische Schlüssel

- Symmetrisches Verfahren mit einem Schlüssel
 - Gleicher Schlüssel zum Verschlüsseln und Entschlüsseln

■ Asymmetrische Schlüssel

- Asymmetrisches Verfahren mit öffentlichem und privatem Schlüssel
 - Verschlüsseln mit öffentlichem Schlüssel
 - Entschlüsseln mit privatem Schlüssel

■ Zertifikat

- Zertifikat eines öffentlichen Schlüssels
 - Bestätigt den Eigentümer und die Eigenschaften eines öffentlichen Schlüssels
 - Enthält den öffentlichen Schlüssel, Name des Eigentümers, Betreff, Gültigkeitsdauer, u.a.

Agenda

- Verschlüsselung
 - Sinn und Zweck
 - Verschlüsselung mit SQL Server
- Verschlüsseln von Daten
 - T-SQL-Befehle
 - Always Encrypted
- Verschlüsselung auf Ebene der SQL Server-Instanz
 - Netzwerkübertragung
 - Datenbanksicherung
- Zusammenfassung und Fazit

Verschlüsselung mit SQL Server

■ SQL Server

- ☐ Verschlüsselungsmethoden
 - Symmetrische Schlüssel
 - Asymmetrische Schlüssel
 - Zertifikate
- ☐ Letztendlich weitere Objekte in einer Datenbank

■ Möglichkeiten

- ☐ Daten mit T-SQL
- ☐ Daten mit »Always Encrypted«
- ☐ Datenbanken mit »Transparent Data Encryption« – TDE
- ☐ Datenbanksicherungen
- ☐ Netzwerkübertragung

Verschlüsselung mit SQL Server / Symmetrische und Asymmetrische Schlüssel

■ Symmetrische Schlüssel

- Empfohlen zum Ver- und Entschlüsseln von Daten
- Bessere Performance als Asymmetrische Schlüssel und Zertifikate
- Anlegen mittels `CREATE SYMMETRIC KEY`
- Schützen des Symmetrischen Schlüssels zwingend erforderlich
- Kennwort, Zertifikat, Asymmetrischer Schlüssel oder Symmetrischer Schlüssel

■ Asymmetrische Schlüssel

- Schlüsselpaar bestehend aus öffentlichem und privatem Schlüssel
- Anlegen mittels `CREATE ASYMMETRIC KEY`
- Empfohlen zum Schützen Symmetrischer Schlüssel
- Schützen des Asymmetrischen Schlüssels erforderlich
- Kennwort oder »Database Master Key«

Verschlüsselung mit SQL Server / Zertifikate

■ Zertifikate

- Erstellen mittels `CREATE CERTIFICATE`
 - Import eines bestehenden Zertifikats in die Datenbank
 - Anlegen eines selbstsignierten Zertifikats – »self-signed Certificate«
- Empfohlen zum Schützen Symmetrischer Schlüssel
- Schützen des privaten Schlüssels vom Zertifikat erforderlich
 - Kennwort oder »Database Master Key«

■ Verwalten von Zertifikaten

- Sichern eines im SQL Server erstellten Zertifikats als Datei empfehlenswert
- Erfordert mittelfristig eine Zertifikatsverwaltung
 - Zugriffsberechtigungen, Überwachen des Ablaufdatums, Einsatzzwecke etc.
 - »Public Key Infrastructure« – PKI

Agenda

- Verschlüsselung
 - Sinn und Zweck
 - Verschlüsselung mit SQL Server
- Verschlüsseln von Daten
 - T-SQL-Befehle
 - Always Encrypted
- Verschlüsselung auf Ebene der SQL Server-Instanz
 - Netzwerkübertragung
 - Datenbanksicherung
- Zusammenfassung und Fazit

Verschlüsseln von Daten / T-SQL

■ Verschlüsseln per T-SQL

- Mehrere T-SQL-Befehle zum Ver- und Entschlüsseln von Daten
 - Symmetrische Schlüssel, Asymmetrische Schlüssel, Zertifikat und Zeichenfolge

■ T-SQL-Befehle

- Ver- und Entschlüsseln per Zeichenfolge
 - ENCRYPTBYPASSPHRASE und DECRYPTBYPASSPHRASE
- Ver- und Entschlüsseln mit Asymmetrischem Schlüssel
 - ENCRYPTBYASYMKEY und DECRYPTBYASYMKEY
- Verschlüsseln mit öffentlichem und Entschlüsseln mit privatem Schlüssel eines Zertifikat
 - ENCRYPTBYCERT und DECRYPTBYCERT
- Ver- und Entschlüsseln mit Symmetrischem Schlüssel
 - ENCRYPTBYKEY und DECRYPTBYKEY

Verschlüsseln von Daten / T-SQL

■ Hinweise zur Verwendung

- Öffnen eines Schlüssels oder Zertifikats mit `OPEN`
- Geöffnet bis zum Ende der Sitzung
- Explizites Schließen durch `CLOSE`

■ Sonderfunktionen

- Zum Entschlüsseln von Daten per Symmetrischem Schlüssel
 - Inklusive automatischem Öffnen des Symmetrischen Schlüssels
- Entschlüsseln eines per Zertifikat geschützten Symmetrischem Schlüssel
 - `DECRYPTBYKEYAUTOCERT`
- Entschlüsseln eines per Asymmetrischem Schlüssel geschützten Symmetrischem Schlüssel
 - `DECRYPTBYKEYAUTOASYMKEY`

Verschlüsseln von Daten / T-SQL

■ Demo

- ☐ Zertifikat anlegen
- ☐ Symmetrischen Schlüssel anlegen und mit Zertifikat schützen
- ☐ Daten per Symmetrischem Schlüssel verschlüsseln und entschlüsseln

Verschlüsseln von Daten / T-SQL / Access und per T-SQL verschlüsselte Daten

■ Access und per T-SQL verschlüsselte Daten

- Kein Ver- und Entschlüsseln der Daten beim direkten Zugriff auf die Daten
 - Eingebundene Tabellen
 - SQL-Anweisungen beim Datenzugriff per DAO mittels Pass Through-Abfragen
 - SQL-Anweisungen beim Datenzugriff per ADO

■ Ver- und Entschlüsseln

- T-SQL-Befehle zum Ver- und Entschlüsseln der Daten erforderlich
- Erweitern der SQL-Anweisungen mit den entsprechenden T-SQL-Befehlen
 - SQL-Anweisungen der Pass Through-Abfragen
 - SQL-Anweisungen der ADO-Zugriffe
- Empfehlung: Verlagern der SQL-Anweisungen in Gespeicherte Prozeduren
 - Ver- und Entschlüsseln der Daten innerhalb der Gespeicherten Prozedur

Verschlüsseln von Daten / Access und per T-SQL verschlüsselte Daten

■ Demo

- ☐ Datenzugriff per eingebundene Tabelle
- ☐ Datenzugriff per Pass Through-Abfrage
- ☐ Datenzugriff per ADO

Verschlüsseln von Daten / T-SQL / Rechtevergabe

■ Rechtevergabe

- ☐ Notwendige Rechte für OPEN
 - VIEW DEFINITION für den Symmetrischen Schlüssel
- ☐ Bei Schutz des Symmetrischen Schlüssels durch ein Zertifikat
 - CONTROL für das verwendete Zertifikat
- ☐ Bei Schutz des Symmetrischen Schlüssels durch einen Asymmetrischen Schlüssel
 - CONTROL für den verwendeten Asymmetrischen Schlüssel
- ☐ Keine explizite Rechtevergabe für CLOSE erforderlich

■ Rechtevergabe bei Sonderfunktionen

- ☐ DECRYPTBYKEYAUTOCERT und DECRYPTBYKEYAUTOASYMKEY
 - VIEW DEFINITION für den Symmetrischen Schlüssel
 - CONTROL für das zum Schutz verwendete Verschlüsselungsobjekt

■ Demo

- ☐ Datenzugriff als Benutzer Hesselbach
- ☐ Notwendige Rechtevergabe

Verschlüsseln von Daten / Verschlüsselungshierarchie

■ Verschlüsselungshierarchie im SQL Server

- Diensthauptschlüssel – »Service Master Key« – SMK
 - Verschlüsselt Informationen der SQL Server-Instanz
 - Anmeldeinformationen, Kennwörter von Verbindungsservern, Datenbankhauptschlüssel
- Datenbankhauptschlüssel – »Database Master Key« – DMK
 - Verschlüsselt Informationen der Datenbanken
 - Private Schlüssel von Asymmetrischen Schlüsseln und Zertifikaten
- Sicherungskopien der Schlüssel auf externen Medien sinnvoll

■ Hinweise

- Feste Bindung verschlüsselter Daten zur Datenbank und zur Instanz
- Hierarchie zum Verschlüsseln nicht zwingend erforderlich
 - Verschlüsselung in vielen Bereichen auch ohne SMK und DMK möglich

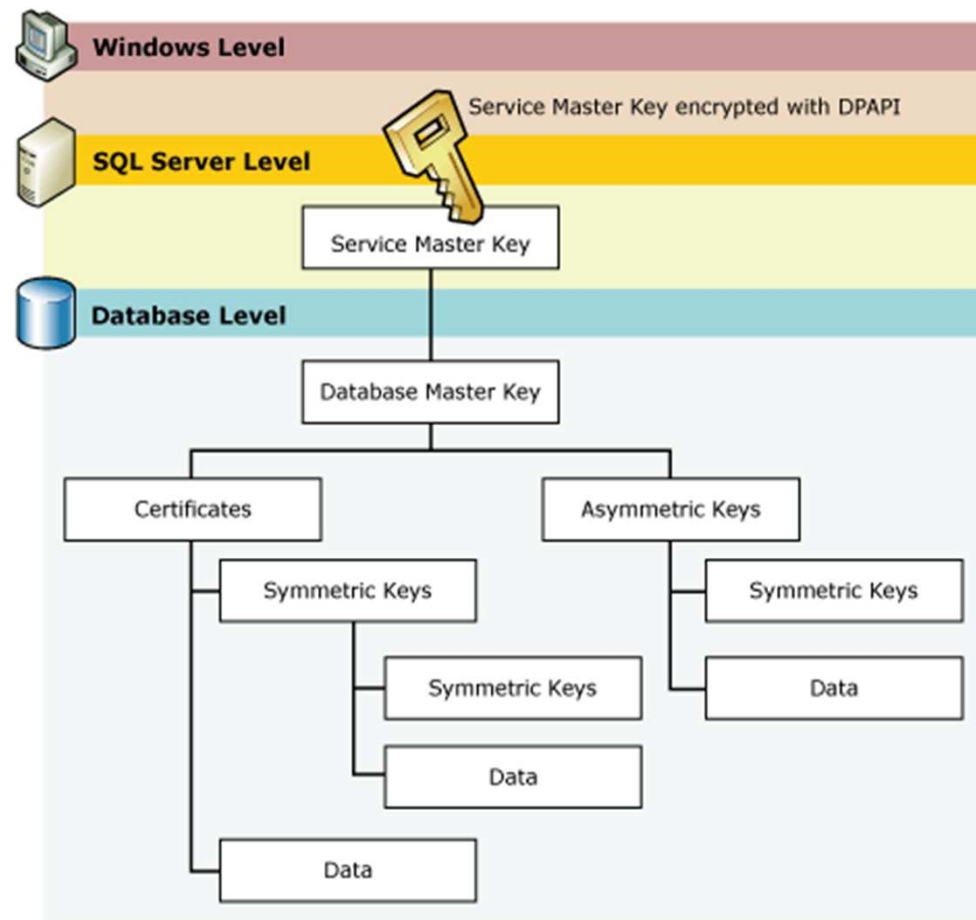
■ Service Master Key – SMK

- Symmetrischer Schlüssel
- Automatisch erstellt beim ersten Start der SQL Server-Instanz
 - Basierend auf den Anmeldeinformationen des Computers und dem Dienstkonto
- Geschützt durch Verschlüsselung per »Windows Data Protection API« – DPAPI

■ Database Master Key – DMK

- Symmetrischer Schlüssel
- Manuelles Erstellen per `CREATE MASTER KEY` in der Datenbank
- Geschützt durch Kennwort
- Erstellt automatisch eine mit Service Master Key verschlüsselte Kopie des DMK
 - Erlaubt das Ver- und Entschlüsseln ohne Angabe des Kennworts vom DMK

Verschlüsseln von Daten / Verschlüsselungshierarchie



Quelle: Microsoft

Verschlüsseln von Daten / Verschlüsselungshierarchie

■ Demo

- ☐ Service Master Key sichern
- ☐ Database Master Key zur Beispieldatenbank anlegen und sichern
- ☐ Zertifikat anlegen und mit Database Master Key schützen
- ☐ Symmetrischen Schlüssel anlegen und mit Zertifikat schützen
- ☐ Daten per Symmetrischem Schlüssel verschlüsseln und entschlüsseln

Agenda

- Verschlüsselung
 - Sinn und Zweck
 - Verschlüsselung mit SQL Server
- Verschlüsseln von Daten
 - T-SQL-Befehle
 - Always Encrypted
- Verschlüsselung auf Ebene der SQL Server-Instanz
 - Datenbanken und Datenbanksicherung
 - Netzwerkübertragung
- Zusammenfassung und Fazit

Verschlüsseln von Daten / Always Encrypted

■ »Always Encrypted«

- Automatisches Ver- und Entschlüsseln von Daten auf Seitenebene
 - Daten einer oder mehrerer Spalten einer Tabelle
- Nur möglich mit Zertifikat aus externem Zertifikatsspeicher
- Erlaubt das Trennen der Datenbesitzer von den Datenverwaltern
 - Lesen und Ändern der Daten nur mit den dafür vorgesehenen Applikationen
 - Verhindert Lesen und Ändern der Daten durch Administratoren

■ Hinweise

- Prüfen der Einschränkungen in Bezug auf Spalten- und SQL Server-Funktionen
- Geschickte Wahl des Benutzers zur Konfiguration und Verschlüsselung
 - Weder IT-Systemadministrator noch Datenbank-Administrator
- Beachten der Besonderheiten und Limitationen beim Datenzugriff

■ Konfiguration

- Erstellen des »Column Master Key« – CMK – mit Verweis auf ein externes Zertifikat
 - Windows-Zertifikatsspeicher, Azure Key Vault oder Schlüsselspeicheranbieter
- Erstellen des »Column Encryption Key« – CEK – mit Verweis auf den CMK
- Auswahl der Spalten und der Verschlüsselungsmethode

■ Verschlüsselungsmethoden

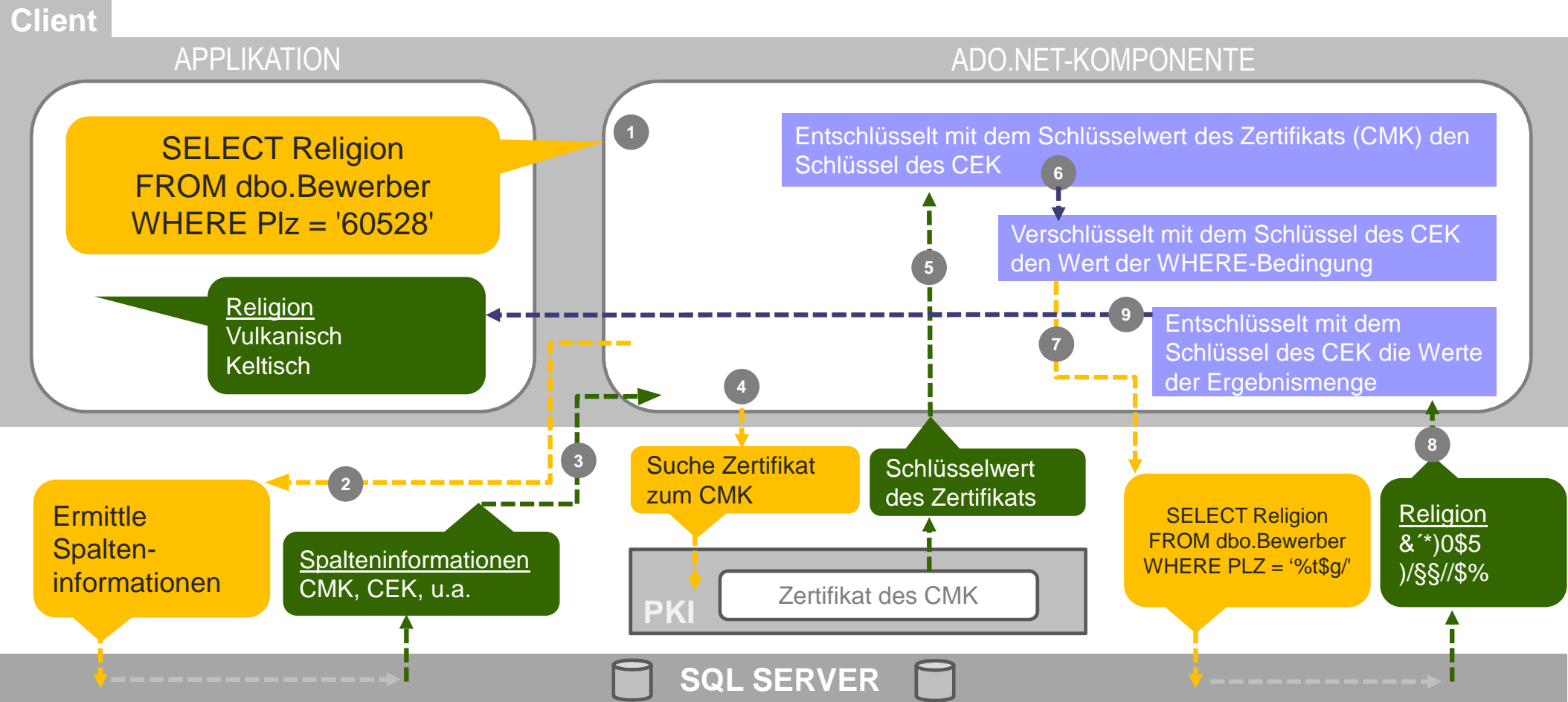
- »Randomized« – Unterschiedlich verschlüsselte Werte zum ursprünglichen Wert
 - Mehr Sicherheit bei weniger Funktionalität
 - Kein Filtern, Gruppieren und Sortieren der Daten möglich
- »Deterministic« – Einheitlich verschlüsselter Wert zum ursprünglichen Wert
- Weniger Sicherheit für mehr Funktionalität
- Filtern mit exakten Werten, Gruppieren und Sortieren möglich

Verschlüsseln von Daten / Always Encrypted

- Demo

- ☐ Konfiguration von Always Encrypted

Verschlüsseln von Daten / Always Encrypted



Verschlüsseln von Daten / Always Encrypted / **Datenzugriff**

■ Limitationen beim Datenzugriff

- Erlaubt SQL-Anweisungen nur in Form parametrisierter Abfragen
- Strenge Regeln zur Codierung von SQL-Anweisungen

■ Regeln beim Datenzugriff

- Angabe von Werten nur per Variable möglich
 - Gilt für Werte in Filterkriterien
 - Gilt für Werte bei `INSERT` und `UPDATE`
 - Gilt für die Übergabe von Werten an die Parameter einer Gespeicherten Prozedur
- Deklaration und Initialisierung einer Variablen in einer Anweisung
 - Typsicherheit bei Variable und Initialwert
 - Keine Systemvariablen zur Initialisierung der Variablen erlaubt
 - Keine Berechnungen zur Initialisierung der Variablen erlaubt

Verschlüsseln von Daten / Always Encrypted / **Datenzugriff**

■ Demo

- ☐ Datenzugriff als Administrator
- ☐ Daten lesen als Benutzer Hesselbach
- ☐ Daten lesen als Admin
- ☐ Limitationen beim Datenzugriff mit Always Encrypted

■ Access und Always Encrypted

- Automatisches Ver- und Entschlüsseln der Daten beim Zugriff auf eingebundene Tabellen
- Benötigt Zugriffsrechte auf das Zertifikat
- Erfordert »ODBC Treiber 13 für SQL Server« und höher
 - Aktivieren der ODBC-Eigenschaft *Spaltenverschlüsselung*
 - Erweitern der Verbindungszeichenfolge mit `ColumnEncryption=Enabled`

■ DAO und ADO mit Always Encrypted

- DAO = Interner Zugriff auf per ODBC eingebundene Tabellen und Pass Through-Abfragen
 - Abhängig von der Konfiguration des ODBC-Treibers und den Limitationen beim Datenzugriff
 - Empfehlung: Aufruf von Gespeicherten Prozeduren in Pass Through-Abfragen
- ADO = Externer Zugriff per OLE DB-Treiber
 - Keine Unterstützung von Always Encrypted in aktuell verfügbaren OLE DB-Treibern

Verschlüsseln von Daten / Always Encrypted / Access und Always Encrypted

■ Demo

- ☐ Datenzugriff per eingebundene Tabelle
- ☐ Datenzugriff per VBA mit DAO
- ☐ Datenzugriff per VBA mit ADO

Agenda

- Verschlüsselung
 - Sinn und Zweck
 - Verschlüsselung mit SQL Server
- Verschlüsseln von Daten
 - T-SQL-Befehle
 - Always Encrypted
- Verschlüsselung auf Ebene der SQL Server-Instanz
 - Datenbanksicherung
 - Netzwerkübertragung
- Zusammenfassung und Fazit

Verschlüsselung auf Ebene der SQL Server-Instanz / Verschlüsselte Datenbanksicherungen

■ Datenbanksicherungen

- Verschlüsseln der Datenbanksicherung empfehlenswert
- Verhindert einfaches Wiederherstellen der Sicherung in fremder SQL Server-Instanz
- Inhalt der Datenbanksicherung nicht mehr lesbar

■ Voraussetzungen

- Zertifikat in Systemdatenbank *master*
- Geschützt durch Database Master Key der Systemdatenbank *master*

■ Konfiguration der Sicherung

- Aktivieren von *Auf neuem Mediensatz sichern und alle vorhanden Sicherungssätze löschen*
- Aktivieren von *Sicherung verschlüsseln*
- Auswahl des Verschlüsselungsalgorithmus und des Zertifikats

Verschlüsselung auf Ebene der SQL Server-Instanz / **Verschlüsselte Datenbanksicherungen**

■ Demo

- ☐ Verschlüsseln von Datenbanksicherungen

Agenda

- Verschlüsselung
 - Sinn und Zweck
 - Verschlüsselung mit SQL Server
- Verschlüsseln von Daten
 - T-SQL-Befehle
 - Always Encrypted
- Verschlüsselung auf Ebene der SQL Server-Instanz
 - Datenbanksicherung
 - Netzwerkübertragung
- Zusammenfassung und Fazit

Verschlüsselung auf Ebene der SQL Server-Instanz / Verschlüsselte Netzwerkübertragung

■ Verschlüsselte Netzwerkübertragung

- Hybrides Verschlüsselungsprotokoll TLS 1.2 – »Transport Level Security«
- Verschlüsselung und Authentifizierung per X.509-Zertifikat
 - CA-Zertifikat oder »self-signed Certificate«

■ Serverseitige Verschlüsselung

- Erzwingen der verschlüsselten Netzwerkübertragung durch SQL Server
- Konfiguration im SQL Server-Konfigurations-Manager

■ Empfehlung

- Verwenden von CA-Zertifikaten
- Aktivieren der clientseitigen Verschlüsselung inklusive Prüfen des Serverzertifikats
 - Verhindert Man-In-The-Middle-Attacken und Downgrade-Attacken

Verschlüsselung auf Ebene der SQL Server-Instanz / **Verschlüsselte Netzwerkübertragung**

- Demo

- ☐ Netzwerkübertragung verschlüsseln

Agenda

- Verschlüsselung
 - Sinn und Zweck
 - Verschlüsselung mit SQL Server
- Verschlüsseln von Daten
 - T-SQL-Befehle
 - Always Encrypted
- Verschlüsselung auf Ebene der SQL Server-Instanz
 - Datenbanksicherung
 - Netzwerkübertragung
- Zusammenfassung und Fazit

Zusammenfassung und Fazit

■ Ver- und Entschlüsseln von Daten

- Per T-SQL und Always Encrypted
- Höhere Datensicherheit beim Datenzugriff
 - Bei T-SQL nur mit zugehörigen Schlüsseln und entsprechenden Rechten möglich
 - Bei Always Encrypted nur mit Rechten auf Zertifikate im Zertifikatsspeicher möglich
- Zusätzlicher Aufwand bei der Datenbankentwicklung
 - Erfordert eigene Konzepte für die Rechtevergabe an den Schlüsseln und Zertifikaten
- Schlechtere Performance beim Datenzugriff

■ Verschlüsseln der Datenbanksicherung

- Verschlüsselte Datenbanksicherung nicht lesbar
- Kein Wiederherstellen in anderen SQL Servern möglich
 - Verschlüsseln der Datenbanksicherung empfehlenswert

Zusammenfassung und Fazit

■ Verschlüsseln der Netzwerkübertragung

- Verschlüsselung und Authentifizierung per Zertifikat
- Verhindert heimliches Mithören bzw. Mitlesen der Datenübertragung
 - Relevant bei Zugriffen über öffentliche Netzwerke
 - Sinnvoll bei firmeninternen Netzwerken

■ Anmerkung

- Steigende Nachfrage nach Verschlüsselung in allen Ebenen
- Ohne Schlüssel kein Entschlüsseln der Daten mehr möglich
 - Es gibt keinen *Schlüsseldienst!*
- Aufbau einer Schlüssel- bzw. Zertifikatsverwaltung mittelfristig unvermeidbar
 - »Public Key Infrastructure« – PKI

Danke

Noch Fragen?

Vielen Dank für die Aufmerksamkeit.