فصل اول

امنیت بهتر

Microsoft SQL Server 2016 سه ویژگی جدید امنیتی را این‌چنین معرفی کرده است: همیشه رمزگذاری شده (Always Encrypted)، امنیت Row-level و ماسک گذاری داده پویا (Dynamic data masking). درحالی‌که همه این ویژگی‌ها مربوط به امنیت هستند، هر کدام نیز سطح مختلفی از حفاظت داده‌ها را در آخرین نسخه از پلتفرم پایگاه داده فراهم می‌کنند. در این فصل، به بررسی موارد استفاده از این ویژگی‌ها، چگونگی عملکرد آن‌ها و زمانی که باید برای محافظت از اطلاعات در پایگاه داده SQL از آن‌ها استفاده شود می‌پردازیم.

همیشه رمزگذاری شده (Always Encrypted)

همیشه رمزگذاری شده، یک فن‌آوری رمزنگاری سمت سرویس‌گیرنده است که در آن داده‌ها نه تنها زمانی که نوشته می‌شوند بلکه زمانی که با یک نرم‌افزار تأیید شده خوانده می‌شوند به‌صورت خودکار رمزگذاری می‌شوند. برخلاف رمزگذاری شفاف داده‌ها که داده‌ها را بر روی دیسک رمزگذاری می‌کند، اجازه می‌دهد تا داده‌ها توسط هر برنامه‌ای که داده‌ها را پرس‌وجو می‌کند خوانده شود، بنابراین همیشه رمزگذاری شده به برنامه سرویس‌گیرنده خود برای رمزگذاری نیاز دارد که موجب برقراری

ارتباط با پایگاه داده می‌گردد. با استفاده از این درایور، برنامه داده‌های رمزگذاری شده را به‌صورت ایمن به پایگاه داده انتقال می‌دهد که پس از آن می‌تواند تنها با یک برنامه کاربردی که دسترسی به کلید رمزنگاری دارد رمزگشایی شود. همچنین هر برنامه پرس‌وجو داده دیگری می‌تواند مقادیر رمزگذاری را بازیابی کند، اما این نرم‌افزار نمی‌تواند داده‌ها را بدون کلید رمزنگاری استفاده کند، در نتیجه ارائه داده بی‌فایده است. به دلیل این معماری رمزگذاری، نمونه SQL Server هیچ‌گاه با نسخه غیر رمز شده از داده‌ها مواجه نمی‌شود. توجه داشته باشید که همیشه رمزگذاری شده فقط فریم ورک دات نت داده را برای SQL Server ارائه می‌کند که نیاز به نصب فریم ورک دات نت نسخه ۴,۶ بر روی کامپیوتر سرویس‌گیرنده و درایور JDBC 6.0 دارد. در این فصل، برای سادگی به هر دو درایور برای سادگی ADO.NET اطلاق خواهد شد.

آغاز به کار با همیشه رمزگذاری شده

برای استفاده از ویژگی همیشه رمزگذاری شده، نیاز به مقدار کمی آماده‌سازی در پایگاه داده برای ذخیره‌سازی جداول رمزگذاری شده خواهید داشت. در حالی‌که این کار را می‌توان با استفاده از یک ویزارد در SQL Server Management Studio انجام داد، استفاده از T-SQL بیشتر توصیه می‌شود؛ بنابراین این فصل بر فرایند پیکربندی T-SQL تمرکز خواهد داشت. آماده‌سازی، یک فرایند دو مرحله‌ای است:

۱. ایجاد تعریف کلید برای ستون اصلی
۲. ایجاد کلید رمزنگاری ستون

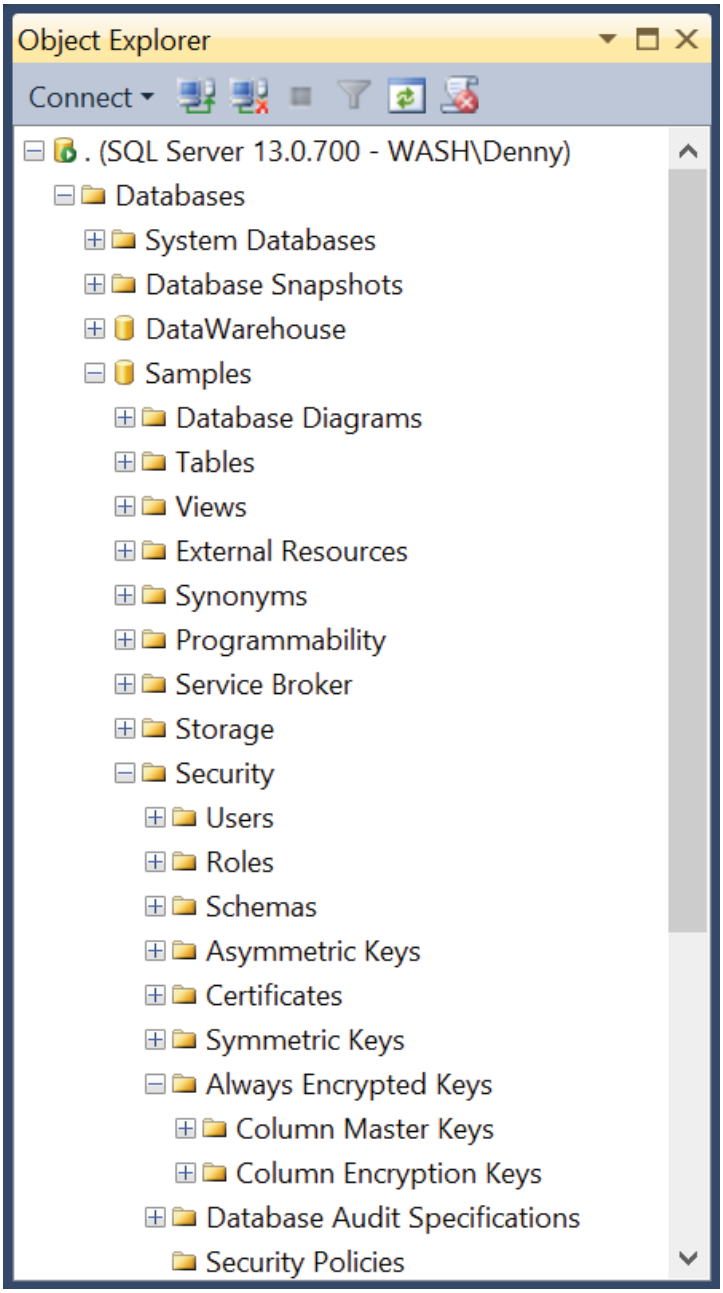
تعریف کلید برای ستون اصلی

کلید اصلی ستون یک گواهی است که در گواهی ویندوز ذخیره شده است، یک ماژول شخص ثالث امنیت سخت‌افزار (HSM) و یا Azure

Key Vault. نرم‌افزاری که داده‌ها را با استفاده از کلید اصلی ستون برای محافظت از کلیدهای مختلف رمزنگاری ستون رمزنگاری می‌کند، مسئول رسیدگی به رمزگذاری داده‌ها در ستون‌های یک جدول پایگاه داده است.

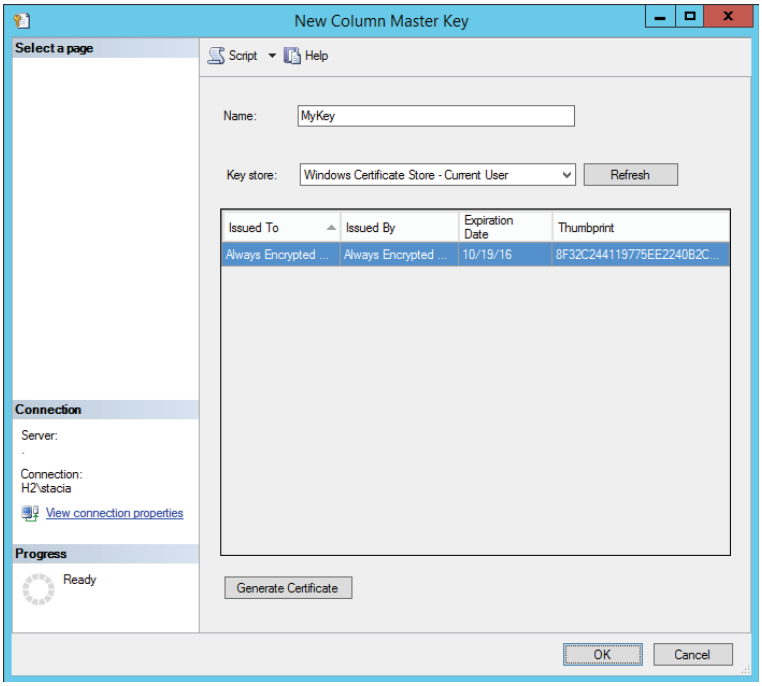
نکته: یک HSM که با عنوان مدیریت سازمانی کلید (EKM) شناخته شده است، نیاز به استفاده از SQL Server Enterprise Edition دارد. در این فصل، استفاده از گواهی self-signed که در Microsoft Certificate Store از سیستم‌عامل ویندوز ذخیره شده است را توصیف می‌کنیم. درحالی‌که این روش پیکربندی بهینه نمی‌باشد، نشان‌دهنده مفاهیم همیشه رمزگذاری شده و قابل انطباق با هر نسخه از SQL Server می‌باشد.

شما می‌توانید یک کلید اصلی ستون با استفاده از SQL Server Management Studio (SSMS) یا با استفاده از T-SQL ایجاد کنید. در SSMS، اتصال به پایگاه داده SQL Server 2016 نمونه‌ای از آن چیزی است که شما می‌خواهید از همیشه رمزگذاری شده برای محافظت از یک جدول پایگاه داده استفاده کنید. در Object Explorer، ابتدا به سمت پایگاه داده، سپس به امنیت حرکت می‌کنیم و در نهایت فایل کلیدهای همیشه رمزگذاری شده را برای نمایش دو زیرمجموعه آن، همان‌طور که در شکل ۱-۱ نشان داده شده است گسترش می‌دهیم.


Object Explorer
Connect
. (SQL Server 13.0.700 - WASH\Denny)
Databases
System Databases
Database Snapshots
DataWarehouse
Samples
Database Diagrams
Tables
Views
External Resources
Synonyms
Programmability
Service Broker
Storage
Security
Users
Roles
Schemas
Asymmetric Keys
Certificates
Symmetric Keys
Always Encrypted Keys
Column Master Keys
Column Encryption Keys
Database Audit Specifications
Security Policies

شکل ۱-۱: فایل کلیدهای همیشه رمزگذاری شده در SQL Server 2016 Object Explorer.

برای ایجاد کلید اصلی ستون، روی فایل کلیدهای ستون اصلی راست کلیک کنید و کلید اصلی ستون جدید را انتخاب کنید. در کادر محاوره‌ای کلید ستون جدید، نامی برای کلید اصلی ستون تایپ کنید. مشخص کنید که آیا ذخیره کلید در کاربر فعلی یا گواهی ماشین محلی یا Azure Key Vault صورت پذیرد و سپس گواهی موجود در لیست را، همانند شکل ۲-۱ انتخاب کنید. اگر هیچ گواهی وجود نداشته باشد و یا اگر می‌خواهید از یک گواهی self-signed جدید استفاده کنید، بر روی دکمه ایجاد گواهی و سپس روی OK کلیک کنید. این مرحله گواهی self-signed را تولید می‌کند و آن را در certificate store حساب کاربر برای اجرا SSMS بارگذاری می‌کند.



شکل ۱-۲: کادر محاوره‌ای کلید اصلی ستون جدید.

نکته: شما باید این مراحل را بر روی یک ماشین قابل اعتماد انجام دهید اما بر روی کامپیوتر میزبان SQL Server نباید قرار بگیرد. به این ترتیب، داده‌های محافظت شده در SQL Server می‌مانند حتی اگر کامپیوتر میزبان به خطر بیفتد.

پس از ایجاد گواهی و پیکربندی آن به عنوان کلید اصلی ستون، شما باید آن را در تمامی کامپیوترهای میزبان مشتریان که نیاز به دسترسی به داده‌ها دارند توزیع کنید. اگر یک برنامه سرویس‌گیرنده، مبتنی بر وب باشد، شما باید بارگذاری گواهی را بر روی وب سرور داشته باشید. اگر یک نرم‌افزار نصب شده بر روی کامپیوتر کاربران است، باید گواهی را برای کامپیوتر هر کاربر به صورت جداگانه صادر کنید.

شما می‌توانید دستورالعمل قابل اجـرا بـرای صـدور گواهینامـه بـرای سیستم‌عامل خود را در آدرس‌های زیر پیدا کنید:

Exporting certificates

 Windows 7 and Windows Server 2008 R2:
https://technet.microsoft.com/en-us/library/cc730988.aspx.

 Windows 8 and Windows Server 2012:
https://technet.microsoft.com/en-us/library/hh848628(v=wps.620).aspx.

 Windows 8.1 and Windows Server 2012 R2:
https://technet.microsoft.com/en-us/library/hh848628(v=wps.630).aspx.

 Windows 10 and Windows Server 2016:
https://technet.microsoft.com/en-us/library/hh848628(v=wps.640).aspx.

Importing certificates

 Windows 7 and Windows Server 2008 R2:
https://technet.microsoft.com/en-us/library/cc754489.aspx.

 Windows 8 and Windows Server 2012:
https://technet.microsoft.com/en-us/library/hh848630(v=wps.620).aspx.
 Windows 8.1 and Windows Server 2012 R2:
https://technet.microsoft.com/en-us/library/hh848630(v=wps.630).aspx.

 Windows 10 and Windows Server 2016: https://technet.microsoft.com/en-us/library/hh848630(v=wps.640).aspx.

مراکز صدور گواهی و حساب خدمات ویژه

هنگامی که شما گواهینامه را در محل مخصوص گواهی بر روی کامپیوترها با برنامه‌ای که داده‌ها را رمزگذاری و رمزگشایی می‌کند، وارد می‌کنید، باید گواهی را در محل گواهی دستگاه در حال اجرای برنامه نیز وارد کنید.

به عنوان جایگزین، شما می‌توانید یک کلید اصلی ستون با استفاده از T-SQL ایجاد کنید. اگرچه ممکن است دریابید که ایجاد کلید نسبت به استفاده از SSMS آسان‌تر است، اسکریپت T-SQL فرآیند تکرار را ارائه می‌دهد تا شما بتوانید به یک سیستم کنترل منبع دسترسی داشته و در صورتی که نیاز به بازسازی سرور داشته باشید امنیت را بررسی کنید. علاوه بر این، به دلیل اینکه بهترین شیوه برای SQL Server 2016، نصب و راه‌اندازی SSMS بر روی کنسول سرور و ویندوز است و همچنین دلایلی چون تامین امنیت نصب و راه‌اندازی گواهی در سیستم‌های ناامن مانند دسکتاپ کاربران، استفاده از اسکریپت T-SQL برای ایجاد کلید اصلی ستون توصیه می‌شود.

برای ایجاد کلید اصلی ستون، از عبارت CREATE COLUME MASTER KEY، همانند مثال ۱-۱ استفاده کنید. در این عبارت شما نیاز به ارائه یک نام برای تعریف، مانند MyKey، همانند مثال نشان داده شده دارید. شما همچنین باید مقدار KEY-STORE-PROVIDER-NAME را به عنوان MSSQL_CERTIFICATE_STORE تنظیم کنید. سپس مسیری برای گواهی به‌عنوان مقدار KEY_PATH مشخص کنید. این مقدار با CurrentUser آغاز می‌شود که از یک

گـــواهی ذخیـــره شـــده در مخـــزن گواهینامـــه حســـاب کـــاربر یـــا LocalMachine استفاده می‌کند. بقیـه مقـادیر یـک رشـته تصـادفی از کاراکترها است که نشان‌دهنده تأثیر گواهی انتخاب شده است. این تـأثیر برای هر تک گواهی منحصر به فرد است.

مثال ۱-۱: ایجاد یک کلید اصلی ستون

```
USE [Samples] GO CREATE COLUMN MASTER
KEY MyKey WITH ( KEY_STORE_PROVIDER_NAME =
N'MSSQL_CERTIFICATE_STORE', KEY_PATH =
N'CurrentUser/My/DE3A770F25EBD6071305B77FB
198D1AE434E6014' ); GO
```

دیگر ارائه‌دهندگان key store

ممکن است از خود بپرسید کدام ارائه‌دهندگان کلید به همراه گـواهی Microsoft SQL Server موجود می‌باشـند. شـما مـی‌توانیـد از میـان چنـدین ارائـه‌دهنـده کلیـد یکـی را انتخـاب نماییـد. یکـی از گزینـه‌هـا MSSQL_CSP_PROVIDER است که اجازه می‌دهد تا شـما از هـر HSM تحت پشتیبانی مایکروسافت CryptoAPI استفاده کنید. گزینـه بعدی MSSQL_CNG_STORE است که اجازه می‌دهد تا شما از هـر HSM تحت پشتیبانی رمزنگاری API استفاده کنید: نسل بعدی. گزینـه سوم، AZURE_KEY_VAULT را به‌عنوان ارائه‌دهنده کلید مشخص می‌کند و نیاز به دانلود و نصـب AZURE_KEY_VAULT بـه‌عنـوان ارائه‌دهنده کلید بر روی ماشین دسترسی به داده‌های حفاظت شـده دارد کـــــه رونـــــد کـــــار در آدرس http://blogs.msdn.com/b/sqlsecurity/archive/2015/11/10/using-the-azure-key-vault-key-store-provider.aspx شرح داده شده اسـت. سـپس

می‌توانید از ارائه‌دهنده سفارشی همان گونه که در http://blogs.msdn.com/b/sqlsecurity/archive"/2015/09/25/creating-an-ad-hoc-always-encrypted-provider-using-azure-key-vault.aspx توصیف شده است استفاده کنید.

اگرچه مقاله بالا یک مثال با استفاده از AZURE_KEY_VAULT ارائه می‌دهد، اما شما می‌توانید برای توسعه ارائه‌دهنده سفارشی نیز اعمال نمایید.

یافتن اثرانگشت گواهی

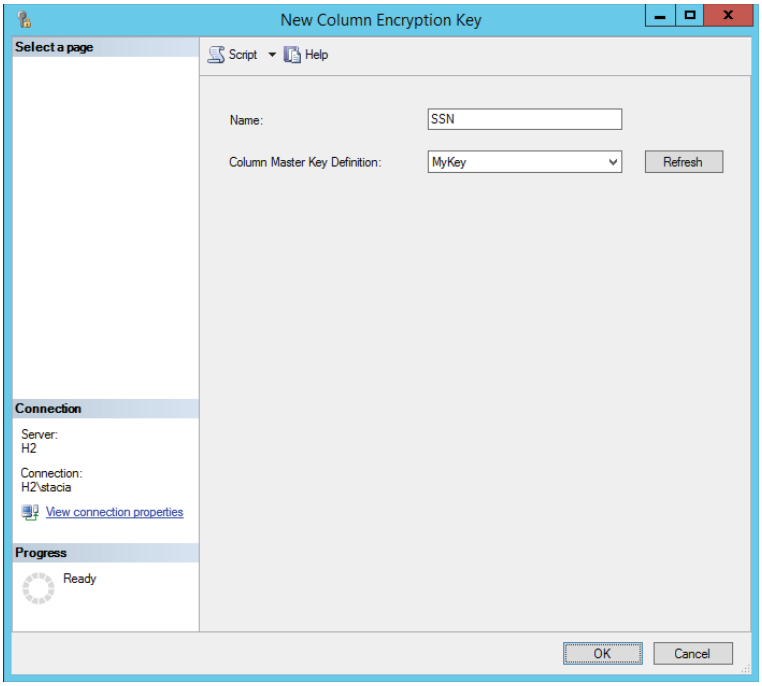
شما به راحتی می‌توانید از اثرانگشت گواهی در درون کنسول مدیریت مایکروسافت (MMC) استفاده کنید. در MMC، منوی File، / Add Remove Snap-In را انتخاب کنید. در کادر باز شده، گواهینامه‌های موجود در لیست در سمت چپ را انتخاب کنید و روی دکمه Add کلیک کنید تا انتخاب شما به سمت راست منتقل شود. گواهینامه‌های موجود در کادر Snap-In این امکان را به شما می‌دهند تا گواهی را انتخاب کنید. همچنین حساب کاربری یا حساب کامپیوتر را بسته به گواهی مورد استفاده‌تان انتخاب کنید. سپس بر روی دکمه Finish کلیک کنید و سپس OK را بزنید. پوشه گواهینامه را برای قرار گرفتن گواهینامه خود در زیرپوشه‌های شخصی / گواهینامه‌ها بسط دهید. گواهینامه را دو بار کلیک کنید، Details Tab را انتخاب کنید و به سمت پایین صفحه بروید تا بتوانید اثرانگشت را که به عنوان مقدار برای ایجاد ستون اصلی کلید استفاده کرده‌اید مشاهده نمایید.

کلیدهای رمزنگاری ستون

پس از ایجاد یک کلید اصلی ستون، شما آماده ایجاد کلیدهای رمزنگاری برای ستون خاص هستید. درایور SQL Server 2016

ADO.NET از کلیدهای رمزنگاری ستون برای به رمز درآوردن داده‌ها قبل از ارسال آن به SQL Server و برای رمزگشایی داده‌ها پس از بازیابی آن از نمونه SQL Server 2016 استفاده می‌کند. همانند کلید اصلی ستون، شما می‌توانید کلیدهای رمزگذاری ستون را با استفاده از T-SQL و یا SSMS ایجاد کنید. درحالی‌که ایجاد کلید اصلی ستون با استفاده از T-SQL ساده‌تر است، ایجاد کلیدهای رمزنگاری ستون با استفاده از SSMS ساده‌تر است.

برای ایجاد یک کلید رمزنگاری ستون، از Object Explorer برای اتصال به پایگاه داده، حرکت به پایگاه داده و سپس امنیت و گسترش پوشه کلیدهای همیشه رمزگذاری شده استفاده می‌کنیم. روی ستون کلیدهای رمزنگاری راست کلیک کنید و سپس کلید رمزگذاری ستون جدیدی را انتخاب کنید. در کادر محاوره‌ای کلید رمزگذاری ستون جدید، یک اسم برای کلید رمزنگاری جدید تایپ کنید و تعریف کلید ستون اصلی را در فهرست کشویی انتخاب کنید و همانند شکل نشان داده شده در ۱-۳، روی OK کلیک کنید. شما هم اکنون می‌توانید کلید رمزنگاری ستون را در تعریف یک جدول جدید استفاده کنید.



شکل ۱-۳: کادر محاوره‌ای کلید رمزگذاری ستون جدید.

برای ایجاد یک کلید رمزنگاری ستون جدید با استفاده از T-SQL، از عبارت CREATE COLUMN ENCRYPTION KEY همانند مثال ۱-۲ استفاده کنید.

مثال ۱-۲: ایجاد کلید رمزنگاری ستون

```
USE [Samples] GO CREATE COLUMN
ENCRYPTION KEY [MyColumnKey] WITH VALUES (
COLUMN MASTER KEY DEFINITION = [MyKey],
ALGORITHM = 'RSA_OAEP', ENCRYPTED_VALUE =
```