عنوان مقاله: آموزش رمزگذاری اطلاعات در SQL Server

نویسنده مقاله: تیم فنی نیک‌آموز

تاریخ انتشار: ۱۴ فروردین ۱۴۰۳

منبع: https://nikamooz.com/data-encryption-in-sql-server/

رمزگذاری اطلاعات در SQL Server از دسترسی غیرمجاز به داده‌های دیتابیس جلوگیری می‌کند و به بهبود امینت آن منجر می‌شود. در مقالات قبلی به بررسی **دستور Select** و **نحوه تعریف محدودیت**، همچنین **دستور Insert** و **عبارت Where** به‌همراه **دستور آپدیت** و **دستور ساخت جدول** و درنهایت، **عملگر Like** و سایر موارد در این RDBMS پرداختیم. حال قصد داریم انواع روش‌های Encryption در SQL Server معرفی کنیم. برای درک بهتر مفاهیم، می‌توانید **مقاله جامع آموزش SQL Server** را نیز مطالعه کنید.

رمزگذاری اطلاعات در SQL Server چیست؟

رمزگذاری پایگاه داده ، روشی امنیتی برای حفاظت از اطلاعات ذخیره‌شده در دیتابیس است. این روش با استفاده از الگوریتم‌های پیچیده، داده‌ها را به رمز تبدیل می‌کند و آن‌ها را برای افراد غیرمجاز به‌صورت غیرقابل خواندن و فهمیدن تبدیل می‌کند. به زبان ساده، تصور کنید دیتابیس SQL Server یک گاوصندوق است که در آن اسناد پرارزشی قرار دارند؛ در این سناریو، رمزگذاری دیتا مانند یک قفل قدرتمند عمل می‌کند و از داده‌های حساس محافظت خواهد کرد.

اهمیت رمزگذاری اطلاعات در SQL Server چیست؟

حال این سؤال پیش می‌آید که اساساً چرا رمزگذاری پایگاه داده حائز اهمیت است؟ در اینجا به برخی از دلایل عمده اشاره می‌کنیم:

- **محافظت از اطلاعات حساس:** رمزنگاری، اطلاعات محرمانه شما شامل اطلاعات مالی، سوابق پزشکی و دیتای شخصی را از دسترس هکرها و افراد غیرمجاز محافظت می‌کند.
- **افزایش سطح امنیت:** رمزگذاری، یک لایه امنیتی اضافی به پایگاه داده شما اضافه می‌کند و به‌عنوان یک سد دفاعی در برابر حملات سایبری عمل خواهد کرد.
- **جلوگیری از سرقت داده‌ها:** درصورت هک شدن پایگاه داده، رمزگذاری مانع از سرقت و سواستفاده از اطلاعات شما توسط هکرها می‌شود.

انواع روش های رمزگذاری داده ها در SQL Server

SQL Server روش‌هایی برای رمزگذاری پایگاه داده ارائه می‌دهد که هر یک، مزیت‌ها و کاربردهای خاص خود را دارا هستند. مهم‌ترین روش‌های رمزگذاری دیتابیس در SQL Server به شرح زیر است:

رمزگذاری شفاف داده ها (Transparent Data Encryption)

این رویکرد به منظور رمزگذاری کل پایگاه داده ، شامل فایل‌های داده و لاگ (log) به کار می‌رود و برای اپلیکیشن‌ها شفاف محسوب می‌شود. در عمل، با به‌کارگیری **روش TDE**، نیازی به تغییر کد برنامه نیست و استفاده از آن، بالاترین سطح امنیت را برای کل پایگاه داده به ارمغان می‌آورد. البته این Encryption فقط در نسخه‌های Enterprise و Standard از SQL Server در دسترس است و همچنین ممکن است بر عملکرد سیستم تأثیر بگذارد. برای آشنایی با ورژن‌های SQL Server، به **مقاله معرفی انواع نسخه های SQL Server و تغییرات آن ها** رجوع کنید.

توجه شود که در TDE Encryption، فرآیند رمزگذاری و رمزگشایی به‌طور خودکار توسط SQL Server انجام خواهد شد و کاربرانی که به پایگاه داده دسترسی پیدا می‌کنند، هیچ‌گونه تفاوتی در عملکرد مشاهده نخواهند کرد. ازسوی دیگر، از آن جایی که فایل‌های گزارش استفاده‌شده برای پشتیبان‌گیری نیز رمزگذاری شده‌اند، Backup های شما نیز محافظت می‌شوند.

به‌صورت کلی، برای فعالسازی TDE در SQL Server ، اقدامات زیر لازم است:

- ایجاد یک Master Key
- ساخت یا دریافت یک گواهی (Certificate) که توسط Master Key محافظت می‌شود.
- ایجاد یک کلید رمزگذاری پایگاه داده که توسط Certificate حفاظت خواهد شد.
- تنظیم دیتابیس برای استفاده از رمزگذاری اطلاعات در SQL Server

در مثال زیر، نحوه رمزگذاری و رمزگشایی در **دیتابیس AdventureWorks2022** ، با استفاده از یک Certificate به نام MyServerCert بررسی می‌شود:

```
USE master;
GO
CREATE MASTER KEY ENCRYPTION BY PASSWORD = '<UseStrongPasswordHere>';
GO
CREATE CERTIFICATE MyServerCert WITH SUBJECT = 'My DEK Certificate';
GO
USE AdventureWorks2022;
GO
CREATE DATABASE ENCRYPTION KEY
WITH ALGORITHM = AES_256
ENCRYPTION BY SERVER CERTIFICATE MyServerCert;
GO
ALTER DATABASE AdventureWorks2022
SET ENCRYPTION ON;
GO
```

عملیات رمزگذاری و رمزگشایی توسط SQL Server و ازطریق رشته‌ها (Threads) پس‌زمینه، زمان‌بندی شده‌اند. برای بررسی وضعیت چنین عملیاتی، می‌توان از Catalog Views یا Dynamic Management Views جدول استفاده کرد.

Always Encrypted

Always Encrypted یا AE، یک ویژگی امنیتی قدرتمند SQL Server است که با استفاده از آن، می‌توان ستون‌های خاصی را در یک جدول پایگاه داده رمزگذاری کرد. در مقایسه با TDE، این رویکرد سطح دقیق‌تری از حفاظت از داده را ارائه می‌دهد. ضمن اینکه Always Encrypted داده‌ها را در سمت کاربر، درون اپلیکیشنی رمزگذاری می‌کند که به پایگاه داده متصل می‌شود. این یعنی، داده‌های حساس قبل از ارسال به سرور رمزگذاری شده و در حالت Rest (ذخیره‌شده) در پایگاه داده باقی می‌مانند. Engine پایگاه داده SQL Server، هرگز داده‌ها را به‌صورت رمزگشایی‌شده، ذخیره یا مشاهده نمی‌کند. این امر به‌طور قابل توجهی خطر دسترسی غیرمجاز به اطلاعات حساس را حتی برای مدیران پایگاه داده (DBA) کاهش می‌دهد. اگر با وظایف و شغل DBA آشنایی ندارید، پیشنهاد می‌شود **مقاله راهنمای مسیر شغلی Database Administrator (DBA)** را مطالعه کنید.

برای راه‌اندازی Always Encrypted در SQL Server، مراحل زیر را دنبال کنید:

- **تهیه کلیدهای رمزگذاری:** در Alway Encrypted از کلیدهایی شامل کلیدهای رمزنگاری ستون (CEK) و کلیدهای اصلی ستون (CMK) استفاده می‌شود. CEK داده‌های درونِ ستون‌های خاصی را رمزگذاری کرده و CMK خود کلیدهای CEK را مانند یک رمز عبور اصلی محافظت می‌کنند. علاوه‌براین، برای ذخیره‌سازی CMK ها به‌صورت امن، باید آن‌ها را خارج از دیتابیس در مکان‌هایی مانند Azure Key Vault ، ماژول امنیتی سخت‌افزار (HSM) و Windows Certificate Store نگهداری کرد.
- **مدیریت کلیدها:** در این گام، لازم است مواردی مانند ایجاد CEK ها، رمزگذاری آن‌ها و ذخیره‌سازی متادیتای کلید انجام شوند. به این ترتیب، یک CEK برای هر ستونی که می‌خواهید رمزگذاری کنید، بسازید. سپس جهت امنیت بیشتر، از CMK به‌منظور رمزگذاری هر CEK استفاده کنید. درنهایت، اطلاعات مربوط به کلیدها را درون پایگاه داده ذخیره کنید. شایان ذکر است که متادیتای CMK، مکان ذخیره‌سازی آن و متادیتای CEK مقدار رمزگذاری‌شده CEK تلقی می‌شود.
- **اعمال رمزگذاری برای ستون‌ها:** ستون‌های خاصی از پایگاه داده که دارای داده‌های حساسی هستند و می‌خواهید رمزگذاری شوند را انتخاب کنید. این کار ممکن است اموری همچون ایجاد جداول جدید با ستون‌های رمزنگاری شده از ابتدا یا رمزنگاری ستون‌های موجود و داده‌های درون آن‌ها را در بر بگیرد.
- **گزینه‌های رمزگذاری اطلاعات در SQL Server:** در این روش Encryption ستون‌ها، می‌توانید الگوریتم رمزگذاری، کلید رمزگذاری ستون و نوع رمزگذاری را تعیین کنید. به‌عنوان مثال، شما می‌توانید برای ستون خاصی الگوریتم AES-256 را به کار ببرید، CEK مدنظر خود را تعیین کرده و انتخاب کنید که رمزگذاری به نوع قطعی یا تصادفی باشد. در نوع رمزگذاری قطعی (Deterministic Encryption)، برای مقدار متن ساده یکسان، همیشه همان مقدار رمزنگاری‌شده ایجاد می‌شود. این روش برای جستجو و Grouping مفید است. هرچند ممکن است برای مجموعه مقادیر کوچک مانند فلگ‌های True / False امنیت کمتری داشته باشد. در رمزگذاری تصادفی، داده‌ها به‌صورت غیرقابل پیش‌بینی Encrypt می‌شوند و به همین دلیل، امنیت افزایش می‌یابد. این رویکرد برای جستجو، Grouping، **ایندکس گذاری** (Indexing) یا جوین روی ستون‌های رمزگذاری‌شده مناسب نیست.

در مجموع، انتخاب یکی از دو روش محافظت از داده ها در SQL Server ، به نیازمندی‌های شما بستگی دارد. توجه کنید که برای رمزگذاری دیتای روی دیسک می‌توانید از Storage Encryption with Encryption Key و همچنین، از Filestream Encryption به منظور ایمن‌سازی داده‌های انتقالی بین اپلیکیشن کلاینت و دیتابیس SQL Server بهره‌مند شد. در بخش بعدی، به این پرسش پاسخ داده می‌شود که چه زمان TDE و چه زمان AE استفاده شوند.

انتخاب روش رمزگذاری اطلاعات در SQL Server

در شرایطی که می‌خواهید تمامی داده‌های پایگاه داده محافظت شوند و همچنین، سادگی پیاده‌سازی و حداقلی بودن تأثیر آن بر کارایی، برایتان اولویت دارد، استفاده از TDE پیشنهاد می‌شود. ازسوی دیگر، اگر ترجیح می‌دهید تنها ستون‌هایی با داده‌های بسیار حساس رمزگذاری شوند و ریسک دسترسی غیرمجاز به آن حداقل شود، لازم است از Always Encrypted یا همان AE استفاده کنید. توجه کنید که هر دوی این روش‌های رمزگذاری اطلاعات در SQL Server ، به استراتژی‌های قدرتمندی برای مدیریت امنیت کلیدهای رمزگذاری نیاز دارند. ضمن اینکه باید با بررسی Trade-off های بالقوه، بار کاری دیتابیس را تجزیه‌وتحلیل کرد تا روش مناسب به کار برود.

مدیریت رمزگذاری اطلاعات در SQL Server

علاوه‌بر رمزگذاری اطلاعات در SQL Server ، باید بتوان با کمک استراتژی‌ها و روش‌هایی از دیتای حساس موجود در پایگاه‌های داده محافظت کرد. اقدامات ضروری برای مدیریت رمزگذاری در SQL Server به شرح زیر است:

مدیریت کلیدها

برای ایمن‌سازی داده‌های رمزگذاری شده در SQL Server ، اعم از TDE و AE، مدیریت دقیق کلید ضروری است. این موضوع، ایجاد کلیدهای رمزگذاری قوی، غیرقابل پیش‌بینی و همچنین، ذخیره‌سازی امن آن‌ها با استفاده از روش‌هایی مانند HSM، برای حداکثر محافظت را شامل می‌شود. علاوه‌براین، تعریف کنترل دسترسی دقیق تضمین می‌کند که فقط پرسنل مجاز می‌توانند کلیدها را مدیریت کنند. درنهایت، داشتن یک برنامه بازیابی کلید، حتی درصورت حذف تصادفی کلید یا خرابی سخت‌افزار، به شما امکان دسترسی به دیتا را می‌دهد.

اعمال سیاست های امنیتی

باید تعیین کنید که کدام داده‌ها نیاز به رمزگذاری دارند و چه روشی برای آن باید اعمال شود. علاوه‌براین، لازم است با نظارت و بررسی مستمر فعالیت‌های رمزگذاری، مطمئن شوید که آن‌ها از سیاست‌های امنیتی تعریف‌شده پیروی می‌کنند یا خیر.

استفاده از ابزارها و راه حل ها

استفاده از متدهای زیر برای رمزگذاری اطلاعات در SQL Server مفید خواهد بود:

- **استفاده از قابلیت‌های Built-In:** می‌توان از قابلیت‌های رمزنگاری پیش‌فرض SQL Server مانند TDE برای کل پایگاه داده و Always Encrypted برای رمزنگاری ستون‌های خاص بهره‌مند شد.
- **استفاده از راهکارهای شخص ثالث:** برای افزایش امکانات و انعطاف‌پذیری، می‌توان از ابزارهای رمزنگاری اضافی کمک گرفت.
- **یکپارچه‌سازی امنیتی:** اطمینان حاصل کنید که مدیریت رمزگذاری داده ها به‌طور یکپارچه با زیرساخت و فرآیندهای امنیتی فعلی موجود ادغام می‌شود.

جمع بندی Encryption :در SQL Server چگونه است؟

محافظت از داده ها در SQL Server به روش‌های مختلفی قابل انجام است که در این مقاله کاربردی‌ترین آن‌ها معرفی شدند. با پیاده‌سازی مؤثر روش‌های رمزگذاری دیتا می‌توان از داده‌های حساس موجود در دیتابیس محافظت کرد. شما می‌توانید با درک دقیق کاربردهای هرروش، بهترین را براساس نیازمندی‌های دیتابیس خود انتخاب کنید.