Encrypting Data with Always Encrypted

Reza Salehi

MCT, MCSA(CLOUD PLATFORM), MCPD

@zaalion linkedin.com/in/rezasalehi2008

Overview

Why encrypt data in Azure SQL Database?

Understanding Azure SQL Database "Always Encrypted"

Deterministic vs. randomized encryption

Demo: Azure SQL Database Always Encrypted

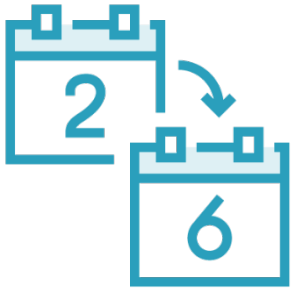
- A new secure column in the *contact* table
- Update *MyAddressBook+* code
- Randomized vs. deterministic

Other Azure SQL encryption options

Module summary

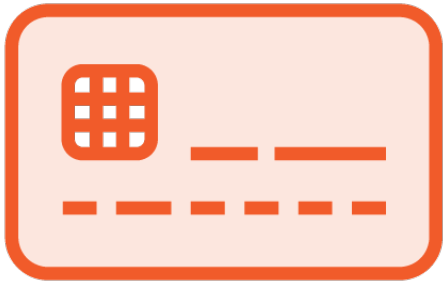
Sensitive Information

Email address



Date of birth

Phone number



Social security number

Salary

Password

Why Encrypt Data in Azure SQL Database?

Bad Guys

Hackers Could access your database files, then they can see confidential information

Good Guys

DBAs have full access over your database. Should they see your information?

Azure SQL Database “Always Encrypted”

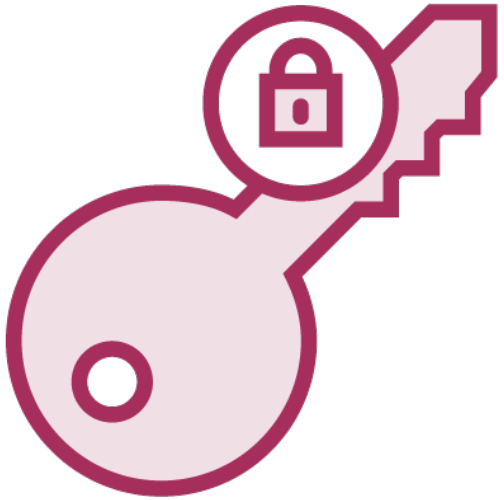
Data encryption technology available in Azure SQL Database and SQL Server

Protect sensitive data at rest on the server

During movement between client and server

Ensuring that sensitive data never appears as plaintext inside the database system

Who Can See the Data Then?



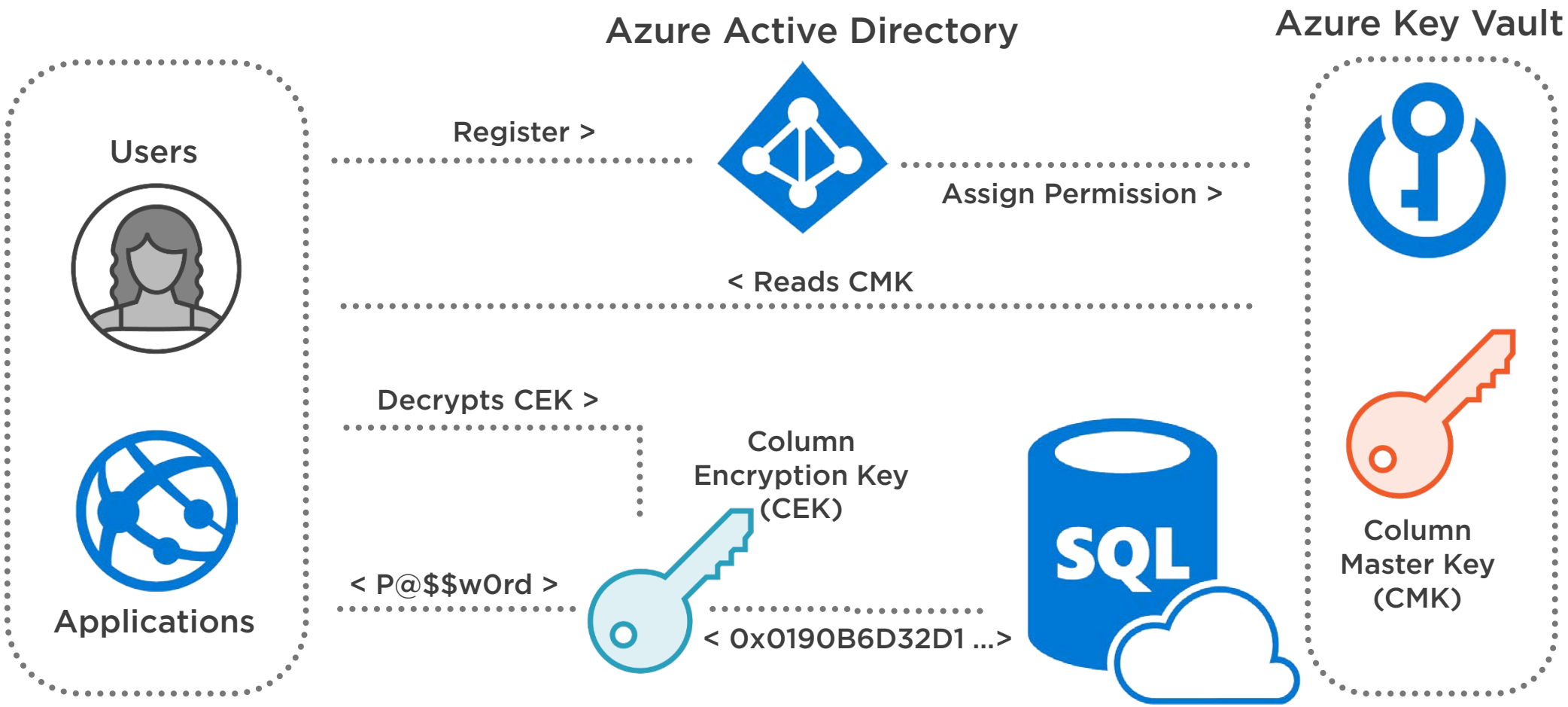
Users/Accounts

Who have the encryption key

Applications/Services

Which have the encryption key

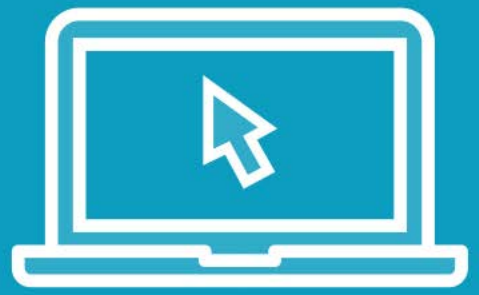
How Does “Always Encrypted” Work?



Column Encryption Types

Randomized	Deterministic
Generates different encrypted value for the same plain text	Generates the same encrypted value for any given plain text
More secure, the encrypted values are difficult to guess	Easy to guess specially for small set of possible encrypted values
Prevents searching, grouping, indexing, and joining on encrypted columns	Allows lookups, equality joins, grouping and indexing on encrypted columns
Confidential comments, not searched	Government ID number, emails, etc.

Demo



Add a SIN number column to the contacts table

Configure Always Encrypted for the new column

- Storing CMK in Azure Key Vault

Update *MyAddressBook+* code to work with the new updates

Randomized vs. deterministic encryption in action

Confirm that *MyAddressBook+* can encrypt and decrypt the data

```
Install-Package
Microsoft.SqlServer.Management.AlwaysEncrypted.AzureKeyVaultProvider

Install-Package Microsoft.IdentityModel.Clients.ActiveDirectory
```

Code Changes for Always Encrypted

Install above *NuGet* packages

```
Column Encryption Setting=Enabled
```

Code Changes for Always Encrypted

Enable “Always Encrypted” in the connection string

```
static void InitializeAzureKeyVaultProvider()

public async static Task<string> GetToken(string authority, string
resource, string scope)
```

Code Changes for Always Encrypted

Register the Azure Key Vault provider with ADO.NET, so the CMK can be read from Key Vault at runtime

```
DynamicParameters parameter = new DynamicParameters();

parameter.Add("@SIN_Number", contact.SIN_Number, DbType.String,
ParameterDirection.Input, 9);
```

Code Changes for Always Encrypted

Use query parameters with fixed length in your queries

Other Azure SQL Encryption Options

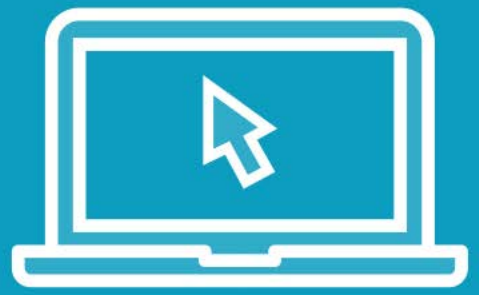
Always Encrypted

Client Side, data is "always encrypted" in transition & in the SQL database

Transparent Data Encryption (TDE)

Server side, encrypts SQL Server, Azure SQL Database, and Azure SQL Data Warehouse data files (at rest)

Demo



Examining Transparent Data Encryption (TDE) option on the server level

Examining Transparent Data Encryption (TDE) option on the database level

Summary

The need to encrypt Azure SQL data

Azure SQL Database Always Encrypted

Deterministic or randomized encryption

Demo: Azure SQL Database Always Encrypted

Other Azure SQL encryption options