DATA() {
EXPOSED;

DATA EXPOSED SPECIAL

Around the Clock with Azure SQL and Azure Data Factory

Americas
February 3, 2021
09:00 - 17:00 PT

Asia
February 4, 2021
09:00 - 17:00 SGT

HOSTED BY
Wee Hyong Tok & Anna Hoffman

16 Sessions | 2 Ask the Expert Panels | 1 Hackathon

DATA() {
EXPOSED;

Securing your Azure SQL Database

Andreas Wolter
Senior Program Manager
SQL Server & Azure SQL Security, Microsoft

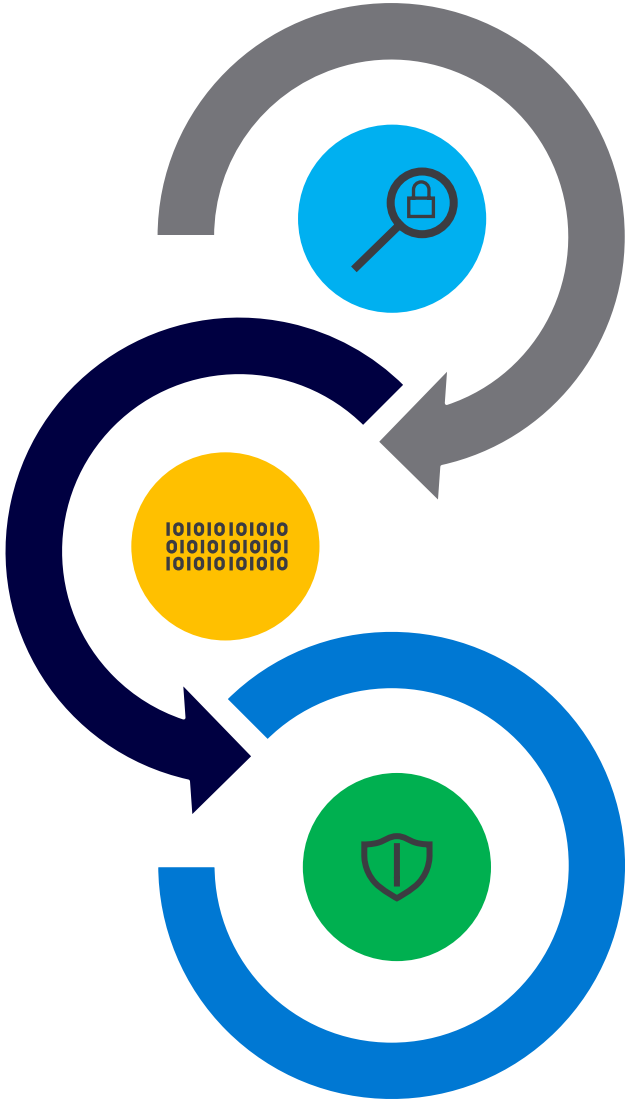
Agenda

- Networking
- Assessment
- Authentication
- Data Protection
- Monitoring & Auditing

Security and Privacy by Design: Data Security Lifecycle

Assess & Classify

- Track configuration and compliance whether it is aligned with the policies.
- Understand where sensitive data lives to identify potential risk and protect confidential information
- Classify data based on content sensitivity, criticality or confidentiality



Implement Protection

- Apply security controls on data based on classification
- Control access to information based on business requirements or need to know basis

Monitor & Respond

- Detect and alert on any unusual user activities:

Securing your databases in Azure SQL (Security Lifecycle)

Secure the Network

IP Based Firewall, Service Endpoints & Vnet Firewall Rules, Private Link, MI specific - NSG, UDR, TLS

Discover and Classify Data

Data Discovery & Classification

Scan for Vulnerabilities

Vulnerability Assessment

Apply Data Protection techniques

Authentication, Roles & Permissions, Encryption

Secure the Application

Row Level Security, Dynamic Data Masking

Monitor security incidents

Auditing, Advanced Threat Protection

Run continuously /on schedule

Every step may need to be repeated in case of new data, users or other needs!

Authentication

Authentication Options

Managed Instance

- Server
  - Special Admin Accounts:
    - Server Admin
    - Active Directory admin
  - **AAD based Logins**
  - SQL Logins
- User Database
  - **Users from AAD Logins**
  - Users from SQL Logins
  - Contained Users from AAD
  - Contained SQL Users

SQL Database

- Server
  - Special Admin Accounts:
    - Server Admin
    - Active Directory admin
  - SQL Logins
- User Database
  - Users from SQL Logins
  - Contained Users from AAD
  - Contained SQL Users

Azure AD Authentication Methods for Azure SQL Database

- With username/password
  - Works for managed and federated domains
  - The easiest way to adopt Azure AD Authentication in existing applications
- Integrated Windows Authentication
  - Works for federated domains and clients on domain-joined machines
  - Eliminates storing password and enables single sign-on

Support Multi-factor authentication (MFA)

- Requires Azure AD Conditional Access enabled (Azure AD P1 and P2)

- Service Principals
  - Supports authentication access using Azure AD applications
  - Requires to use a secret or certificate
- Token-based Authentication
  - Gives application full control over access token acquisition
  - Enables authentication using certificates

Data Protection techniques in Azure SQL

Access Control in Azure SQL

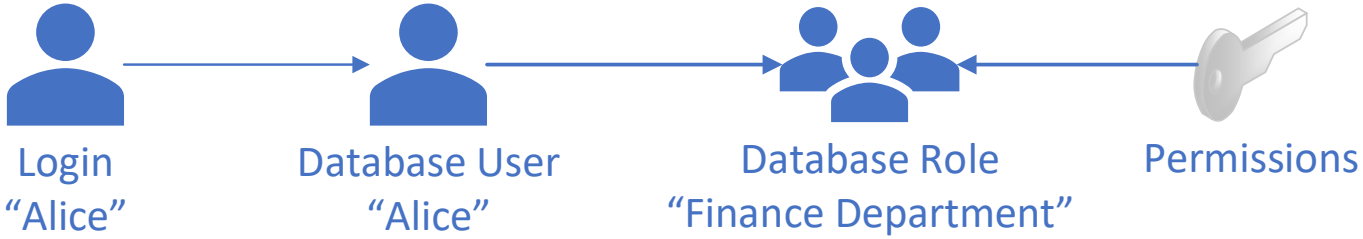
- Permissions and Schemas
  - Use the Principle of Least Privilege when granting permissions
    - **Azure SQL comes with up to 240 permissions depending on the SKU**
  - Inside the databases, **schemas** can be used to group objects with similar security requirements
- Roles
  - Roles group permissions together and make security manageable
  - There are built-in roles and you can create custom roles that have permissions that suit your needs

➢ Add Users to Roles, Grant Permissions to Roles - on schema-level if possible

URP or LURP

SQL Database

Managed Instance



Encryption

- Encryption further strengthens security and can prevent data exfiltration
- **TLS** (Transport Layer Security) applies Encryption-**in-flight**
- **TDE** protects data **at rest**
  - On by default in Azure since 2017
- **Always Encrypted** protects data **in use** from high privileged users by applying Separation of Duties between data owner and data manager (like a DBA)

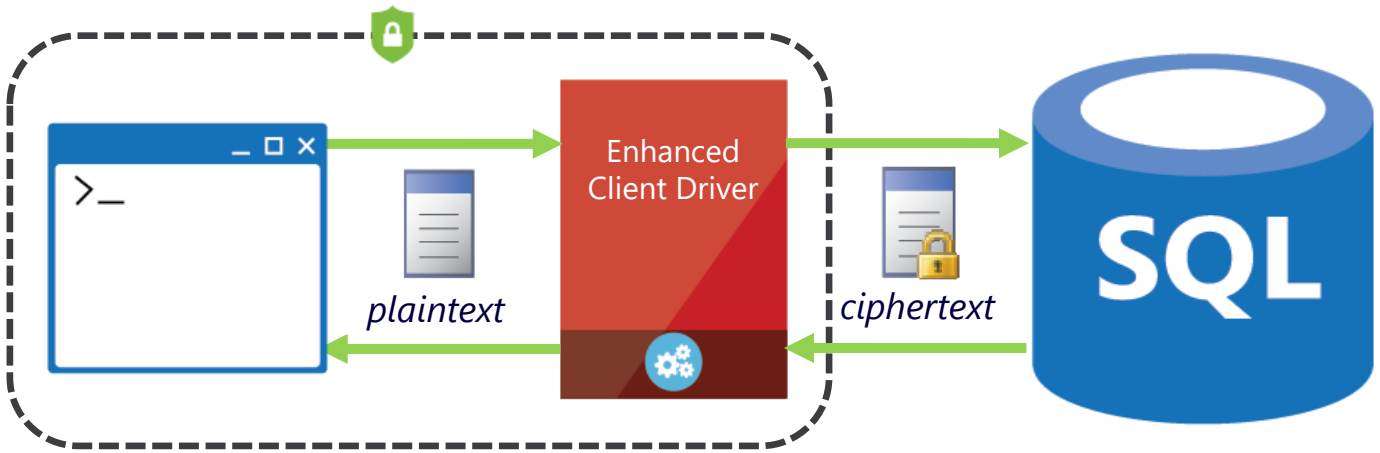
TDE

- Encryption at-rest
  - Data encrypted a symmetric data encryption key (DEK) using AES-256
  - The DEK encrypted with a TDE Protector – an asymmetric key
- TDE with service-managed keys (SMKs)
  - On by default
  - A TDE Protector is a certificate unique for each server
  - Automatically rotated by Microsoft
- TDE with customer-managed keys (CMKs)
  - The TDE protect is a customer-owned asymmetric key stored in Azure Key Vault

Azure SQL DB and SQL Server 2016 and higher

Always Encrypted

Protects sensitive data **in use** from high-privileged yet unauthorized SQL users both on-premises and in the cloud



Client side Encryption

Client-side encryption of sensitive data using keys
that are ***never*** given to the database system

Encryption Transparency

Client driver transparently encrypts query parameters and decrypts encrypted results

Queries on Encrypted Data

Support for equality comparison, including join, group by and distinct operators via deterministic encryption

How Always Encrypted works

Client

```
using (SqlCommand cmd = new SqlCommand(
"SELECT Name FROM Patients WHERE SSN =
@SSN“
 , conn))
{
 cmd.Parameters.Add(new SqlParameter(
  "@SSN", SqlDbType.VarChar, 11).Value =
   "111-22-3333");
 SqlDataReader reader =
  cmd.ExecuteReader();
}
```

Result set (plaintext)

Name
John Smith

CMK Store

CMK

CEK

Enhanced Client Driver

SQL Server or Azure SQL Database

```
exec sp_describe_parameter_encryption
@params = N'@SSN VARCHAR(11)'
, @tsql = N'SELECT Name FROM Patients WHERE SSN =
@SSN'
```

Param	Encrypted CEK Value	CMK Store Provider Name	CMK Path
@SSN		AZURE_KEY_VAULT	https://my.vault.azure.net:443/keys/CMK1/b94d985

```
EXEC sp_execute_sql
N'SELECT Name FROM Patients WHERE SSN = @SSN'
, @params = N'@SSN VARCHAR(11)', @SSN=0x7ff6a54ae6d
```

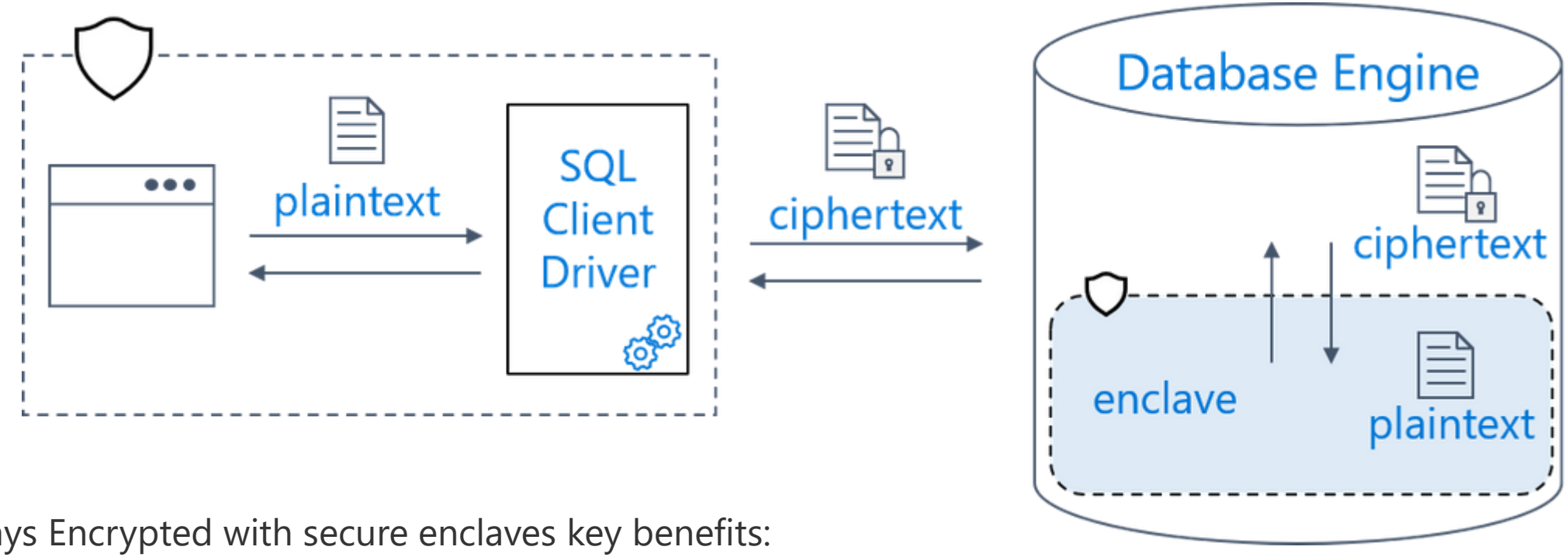
Param	Encrypted CEK Value	CMK Store Provider Name	CMK Path
@Name		AZURE_KEY_VAULT	https://my.vault.azure.net:443/keys/CMK1/b94d985

Name
0x19cae706fbd9

Patients table

Name	SSN
0x19cae7...	0x7ff6a54...
0xfbd9ae...	0x654ae6...

Always Encrypted with secure enclaves in Azure SQL Database preview



Always Encrypted with secure enclaves key benefits:

•**Rich confidential queries**, including pattern matching (LIKE) and range comparisons. These new capabilities make it possible to protect a much broader set of sensitive information (names, address, phone numbers, sensitive numerical data) without painful compromises.

•**In-place encryption** – allowing cryptographic operations inside the secure enclave, to eliminate the need to move the data outside of the database for initial encryption or key rotation.

Call to Action

- **Networking**
- Use Private Link and set Deny Public Network Access to Yes
- Enforce TLS 1.2 on the SQL Database server
- **Assessment**
- Run Data Discovery after Schema-additions
- Configure VA rules with baselines and run VA on schedule (i.e. weekly)
- **Authentication**
- Use integrated Auth or token (for App scenarios) whenever possible
- **Data Protection**
- Use roles to assign permissions
- Encrypt sensitive data with Always Encrypted and role Separation
- **Auditing & Monitoring**
- Configure Auditing for at least any security-impacting operation (such as creating Users, changing permissions)
- Turn on ATP (bundled with VA) on all production servers

Further reading

Introduction into security principles in the context of database systems
https://aka.ms/SecurityPrinciplesSQL

Schema-design for SQL Server: recommendations for Schema design with security in mind
http://andreas-wolter.com/en/schema-design-for-sql-server-recommendations-for-schema-design-with-security-in-mind/

Always Encrypted
https://docs.microsoft.com/en-us/sql/relational-databases/security/encryption/always-encrypted-database-engine

Always Encrypted with secure enclaves
https://docs.microsoft.com/en-us/sql/relational-databases/security/encryption/always-encrypted-enclaves

Check out our "Playbook for addressing common security requirements with Azure SQL Database and Azure SQL Managed Instance": https://aka.ms/AzureSQLDBSecurityPlaybook

DATA() {
EXPOSED;

Learn with us!

View our on-demand playlist:
aka.ms/azuresqlandadf

@AzureSQL
@AzDataFactory

Microsoft Azure