Database Security

Semester 2 / 1400-1401 (Spring 1401)

Instructor: Dr. Fakhrahmad

TA: Rahmani

Homework #2: Data Encryption in SQL Server

The learning objective of this homework is for you to gain hands-on experience with data encryption in database, especially Microsoft SQL Server.

Database encryption is the process of converting data, within a database, in plain text format into a meaningless cipher text by means of a suitable algorithm. The database encryption protects the stored data. Database encryption is done to encrypt sensitive data like credit card numbers, medical records, etc. on the tables, columns, or rows of a database.

In this homework you will learn two techniques and technologies available for database encryption:

1. Column Encryption (Cell Level)
2. Transparent Data Encryption (TDE)

Column Encryption

Today (almost) every organization has at least one application built on Microsoft's SQL Server database. it is also impossible to run a business without handling sensitive information and storing data such as customer names, credit card numbers, bank account numbers, passwords, email addresses, or other personally identifiable information (PII) or private health information (PHI) in your SQL Server database. With such data stored in tables, you have a few options to protect data. First, you can protect the data using *views*. Second, we can also *assign column level permissions to users*. Are there any other options available?

Cell level encryption let us to encrypt a column which contains critical information like credit card numbers.

TDE:

Transparent Data Encryption (TDE) is a feature introduced in SQL Server 2008 and available in later versions for bulk encryption at the database file level (data file, log file and backup file) i.e. the entire database at rest. Once enabled for a database, this feature encrypts data into pages before it is written to the disk and decrypts when read from the disk. The best part of this feature is, as its name implies, it's completely transparent to your application. This means literally no application code changes (only administrative change to enable it for a database) are required and hence no impact on the application code\functionalities when enabling TDE on a database being referenced by that application.

Assignment

Column Encryption: for this section you will walk through the processes of encrypting a column in a table which contains credit card information of customers of Shirazu's company by using SQL Server symmetric key encryption. Please be informed that encrypted data must be stored as **varbinary** and then recast back to the appropriate data type when read.

You should encrypt the "CCNumber" field of the database. For creating and filling the considered table, please use the provided script (after creating a DB named ShirazuCompany). Please be informed that the "customersData.csv" file is attached and you should change the "'E:\Courses\DBS\Encryption\customersData.csv'" address suitably based on your file system.

You should use **AES** for encrypting the mentioned column. Once that you have encrypted the column, you should be also able to do decryption.

Hint: you first should add another field to the table with type **varbinary** and then populate it with the encrypted values of "CCNumber" column.

```
USE ShirazuCompany;
GO
CREATE TABLE Customers(
[CustomerNumber] int NOT NULL,
[FirstName] [varchar](50) NULL,
[LastName] [varchar](50) NULL,
[Phone] [varchar](50) NULL,
[Address] [varchar](50) NULL,
[City] [varchar](50) NULL,
[State] [varchar](50) NULL,
[Zip] [varchar](10) NULL,
[Email] [varchar](50) NULL,
[Birthdate] [varchar](50) NULL,
[Anniversary] [varchar](50) NULL,
[CCNumber] [varchar](20) NULL,
CONSTRAINT [PK_Customers] PRIMARY KEY CLUSTERED
(
[CustomerNumber] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
/******************************/
BULK
INSERT Customers
FROM 'E:\Courses\DBS\Encryption\customersData.csv'
WITH
(
FIELDTERMINATOR = ',',
ROWTERMINATOR = '\n'
)
GO
```

TDE: for this section, you should enable the TDE for the ShirazuCompany database using AES_128 Algorithm. You can use this command to know that your query works correctly. If the result of the encryption_state be number *3*, it is ok.

```
SELECT DB_NAME(database_id), encryption_state
FROM sys.dm_database_encryption_keys
```

Submission

For column encryption, you should provide the modified table and your queries which you have done the encryption and decryption with them.

For TDE, you should provide the query which enables TDE on SQL Server for ShirazuCompany database.

You should also provide a self-descriptive report of the whole process (cell level and TDE).

Please email your solutions till May 20th, 2020 (1401/02/30) to peyman.rahmani@gmail.com with a subject of the following format: DBS.HW#2.[Your Student ID].[Your full name].

If you have any questions, please feel free to contact me by email at peyman.rahmani@gmail.com.

Kind regards,

DB Sec. Team