

## Lab: Data Encryption

- This is worth 2 points.
- The due date is tomorrow midnight.
- Use the following naming convention: homework, underscore, last name, first initial, and extension (e.g., Lab\_Encrypt\_ImG.docx).

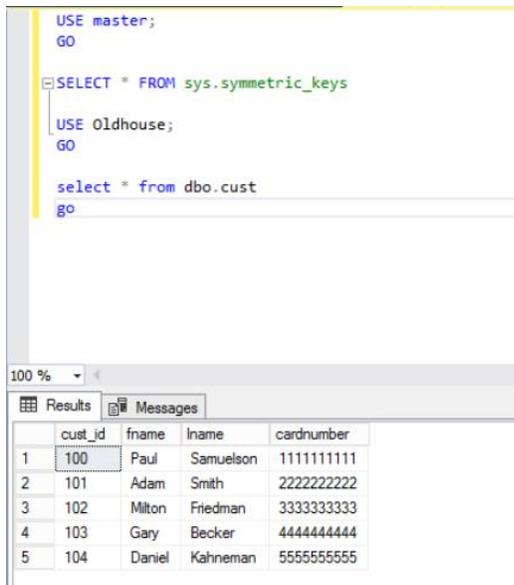
### 1. Preparation

First, if your SQL Server does not have Oldhouse database, create it using this script: **Oldhouse-Table-Crete (Lab).sql**.

Next, perform the lab using this script: **Encryption-Cert (Lab).sql**.

### 2. Deliverables

```
-- Display the original table
select * from dbo.cust
go
/* Task #1: Show the original table in a screen shot. */
```



```
-- Display the encrypted table
select * from dbo.cust_encrypt
go
/* Task #2: Show the encrypted table in a screen shot. Also, explain why we need to
change the data type for encryption. */
```

```

go
declare @passphrase varchar(128)
set @passphrase = 'unencrypted credit card numbers are bad, um-kay'
insert dbo.cust_encrypt
(
    fname,
    lname,
    cardnumber_encrypt
)
select
    fname,
    lname,
    cardnumber_encrypt = EncryptByPassPhrase(@passphrase, cardnumber)
from dbo.cust

```

fname	lname	cardnumber_encrypt
Paul	Samuelson	0x01000000B1488F3B4B086245BB4EA0849BCF6DCFAEBCED...
Adam	Smith	0x01000000F04E87BE120AF40BA78F575CD583F354BA5C8FA...
Milton	Friedman	0x01000000F3BD0D52429643A3A06E475A95B511652BD4137...
Gary	Becker	0x01000000293C748A0805257B2321E66AFD58D9BBAA67232...
Daniel	Kahneman	0x0100000076EF016772653A1F5C0B4D8E7466C0F232316D96...

We change the data type for encryption because the data needs to be protected from those who have a chance to enter in the database to take sensitive information.

```

-- Display the encrypted table
select * from dbo.cust_encrypt
go
/* Task #3: Show the encrypted table in a screen shot. Also, explain the encryption
process after Task #2. */

```

```

insert dbo.cust_encrypt (
    fname,
    lname,
    cardnumber_encrypt
)
select
    fname,
    lname,
    cardnumber_encrypt = EncryptByKey(KEY_GUID('BillingSymKey'), cardnumber)
from dbo.cust
select * from dbo.cust_encrypt
go

```

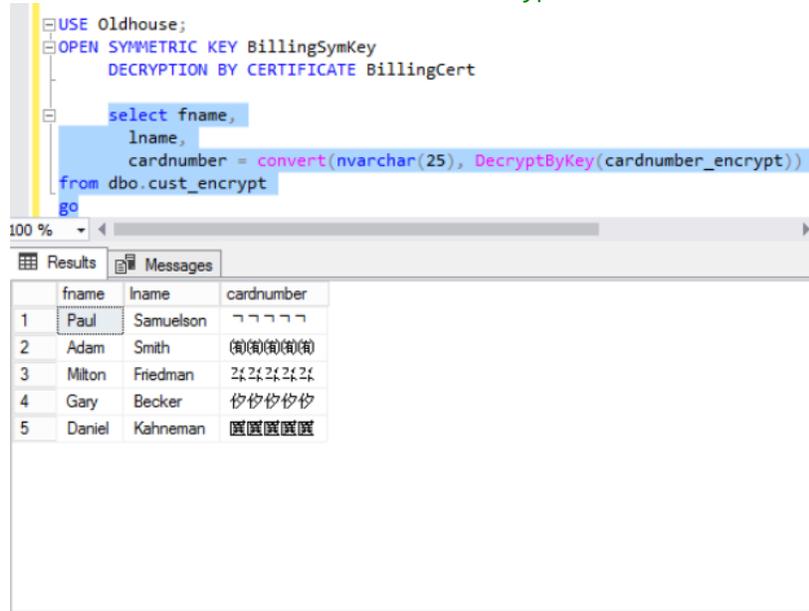
fname	lname	cardnumber_encrypt
Paul	Samuelson	0x005B0096AB21CE4D95252AD6B3BBBBD9010000017A1915...
Adam	Smith	0x005B0096AB21CE4D95252AD6B3BBBBD90100000176DB8...
Milton	Friedman	0x005B0096AB21CE4D95252AD6B3BBBBD901000000B73B827...
Gary	Becker	0x005B0096AB21CE4D95252AD6B3BBBBD901000000ECA069...
Daniel	Kahneman	0x005B0096AB21CE4D95252AD6B3BBBBD9010000007289B4...

The encryption process after task 2 is that we make a certificate using the database, and then creating a symmetric key for it. Then it becomes encrypted by using the certificate. Then the table is emptied by truncating it. Then the table key is decrypted using the certificate, and then the rows of the table are inserted by using the key encrypted by the Certificate.

```

-- Display the decrypted table
select fname,
       lname,
       cardnumber = convert(nvarchar(25), DecryptByKey(cardnumber_encrypt))
from dbo.cust_encrypt
go
/* Task #4: Show the encrypted table in a screen shot. Also, explain the decryption
process after Task #3. */
/* Did you get the original data back? If not, what's wrong? */
/* Hint: Check out the current data type of cardnumber with the original one */

```



The decryption process is started by selecting the Oldhouse database. Then the symmetric key is opened and used by the certificate for decryption. Once the certificate is used for decryption, the user can then display the table that has been decrypted. I did not get the original data back because the decrypted tables, data type is cardnumber\_encrypt which still protects the data. In the original table, the data type for that is a varchar which can be displayed when selected. So, if the decrypted table does not have the correct data type, the data can't be shown in its original form.