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-Create (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. */
    USE master;
    GO
   SELECT * FROM sys.symmetric_keys
    USE Oldhouse;
    GO
    select * from dbo.cust
    go
100 % - <
 I Results Messages
     cust_id fname Iname
                          cardnumber
          Paul
 1
    100
                 Samuelson 1111111111
 2
           Adam Smith
     101
                          2222222222
 3
     102
           Milton Friedman
                          33333333333
 4
     103
                          444444444
           Gary
                  Becker
                          5555555555
 5
     104
            Daniel
                 Kahneman
```

```
-- 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. $\ast/$

	go			^						
E	<pre>E declare @passphrase varchar(128) set @passphrase = 'unencrypted credit card numbers are bad, um-kay' insert dbo.cust_encrypt</pre>									
		fname, lname, cardnumbe	r_encrypt							
) select	fname, lname, cardnumbe	r_encrypt = EncryptByPassPhrase(@passphrase, car	dnumber)						
100 %	from d	bo.cust								
100 A	Results	Messages								
	fname	Iname	cardnumber_encrypt							
1	Paul	Samuelson	0x0100000B1488F3B4B086245BB4EA0849BCF6DCFAEBCED							
2	Adam	Smith	0x01000000F04E87BE120AF40BA78F575CD583F354BA5C8FA							
3	Milton	Friedman	0x01000000F3BD0D52429643A3A06E475A95B511652BD4137							
4	Gary	Becker	0x0100000293C748A0805257B2321E66AFD58D9BBAA67232							
5	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. */
```

	- ir	sect dbo c	ust encount (T	
ľ	1	fname.	usc_encrypt (()	1
		Iname,			
		cardnumbe	r encrypt		
)	-		
	select				
		fname,			
		lname,			
	2000	cardnumbe	<pre>r_encrypt = EncryptByKey(KEY_GUID('BillingSymKey</pre>),cardnumber	
	Trom c	ibo, cust			
	-	1 1 1 1 1			
	select	from db	o.cust encrypt		
	select	: * from db	o.cust_encrypt		
	select go	: from db	o.cust_encrypt	-	
	go	: * from db	o.cust_encrypt		
	go	: * from db	o.cust_encrypt		
%	go	: * from db	o.cust_encrypt	•	
%	select go 6 - 4 Results	: * from db	o.cust_encrypt	×	
%	select go & • 4 Results fname	E from db E Messages Iname	o.cust_encrypt cardnumber_encrypt		
%	select go Results fname Paul	Messages Iname Samuelson	o.cust_encrypt cardnumber_encrypt 0x005B0096AB21CE4D95252AD6B3BBBBD90100000017A1915		
34	select go Results fname Paul Adam	Messages Iname Samuelson Smith	o.cust_encrypt cardnumber_encrypt 0x005B0096AB21CE4D95252AD6B3BBBBD90100000017A1915 0x005B0096AB21CE4D95252AD6B3BBBBD901000000176DB8		
74	select go Results fname Paul Adam Milton	Messages Iname Samuelson Smith Friedman	o.cust_encrypt cardnumber_encrypt 0x005B0096AB21CE4D95252AD6B3BBBBD90100000017A1915 0x005B0096AB21CE4D95252AD6B3BBBBD901000000176DB8 0x005B0096AB21CE4D95252AD6B3BBBBD901000000B73B827	, . , .	
%	select go Results fname Paul Adam Milton Gary	Messages Iname Samuelson Smith Friedman Becker	o.cust_encrypt cardnumber_encrypt 0x005B0096AB21CE4D95252AD6B3BBBBD90100000017A1915 0x005B0096AB21CE4D95252AD6B3BBBBD901000000176DB8 0x005B0096AB21CE4D95252AD6B3BBBBD901000000873B827 0x005B0096AB21CE4D95252AD6B3BBBBD901000000ECA069		

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.

Disp	lay the	decrypte	d table								
select	fname,										
	lname	و									
	cardn	umber =	<pre>convert(nvarchar(25), DecryptByKey(d)</pre>	cardnumber_encrypt))							
<pre>from dbo.cust_encrypt</pre>											
go											
/* Task	#4: Sho	w the en	crypted table in a screen shot. Also	o, explain the decryption	I						
process	after T	ask #3.	*/								
/* Did	you get	the orig	inal data back? If not, what's wrong	g?	*/						
/* Hint	: Check	out the	current data type of cardnumber with	h the original one	*/						
	ldhouse;										
OPEN	SYMMETRIC K	EY BillingSy	ymKey								
-	DECRYPTION	BY CERTIFIC	Ale BillingCert								
E.	select fnam	e,									
	lname,										
from	cardnumbe dho cust en	r = convert	<pre>(nvarchar(25), DecryptByKey(cardnumber_encrypt))</pre>								
go	abo.cusc_cn	crype									
100 % 👻 🖣			►								
I Results	Messages]									
fname	Iname	cardnumber									
1 Paul	Samuelson										
2 Adam	Smith	御御御御御									
3 Milton	Friedman	2626262626									
4 Gary	Becker	伦伦伦伦伦									
5 Daniel	Kahneman	展展展展展									
L				-							

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.