**Changing Certificates in Transparent Data Encryption**

v.1

I Original Draft…………………………………………………………………………..Tim Collins 3/16/2016

**Introduction and assumptions**

This SOP assumes you have a database currently encrypted and, want to change keys and certificates. Once this is done, it is assumed the user will want to back up the database in this state and, then change the key and certificate again to the original state.

There are roles and permissions that might need to be granted to the user performing these operations. Specifically, you may need:

* grant control on certificate::MyCertificateName to [washdc\User123]
* grant view definition on certificate::MS\_DatabaseMasterKey to [washdc\User123]
* grant alter any symmetric key to [washdc\User123]

\*\*Note that “User123” and “MyCertificateName” are only place holders.

It should be noted that all paths, passwords, and any other sensitive information in this document is purely fictional and will have to be modified by the user performing the operations.

**Contents**

**1 – Source**

1a – Identify current cert/key

1b – Backup current cert/key

1c – Create New cert

1d – Backup New cert

1e – Alter Database Encryption & Drop current Key

1f – Associate Database with new cert & set encryption

1g – Backup Database

1h – Move Cert, Keys, Backup to Destination

1i – Alter Database Encryption & Drop current Key

1j - Associate Database with new cert & set encryption

1k – Review encryption status

**2 – Destination**

2a – Determine if there is already a Master Key

2b – Create Master if one doesn’t exist

2c – Create Cert/Key from Source files

3d – Restore Database

3 e – Review encryption status

**Begin Source Process**

**1 – Source**

1a – Identify current cert/key

These scripts will help you determine what keys and certificates are on the database. Be sure to run this in the Master as well as in the Source Database so you can see the Master Key as well. Note that an encryption state of 3 is encrypted.

select \* FROM sys.dm\_database\_encryption\_keys

select \* from sys.symmetric\_keys

select \* from.sys.certificates

select \* from sys.databases

--This provides the cert name and database

select

database\_name = d.name,

dek.key\_algorithm,

cert\_name = c.name

from sys.dm\_database\_encryption\_keys dek

left join sys.certificates c

on dek.encryptor\_thumbprint = c.thumbprint

inner join sys.databases d

on dek.database\_id = d.database\_id;

1b – Backup current cert/key

Use Master

go

BACKUP CERTIFICATE CurrentCert TO FILE = 'd:\TDE\CertName.cer'

WITH PRIVATE KEY ( FILE = 'd:\TDE\KeyName\_Key.pvk' ,

ENCRYPTION BY PASSWORD = 'AStrongPassword' );

GO

1c – Create New cert

USE Master

GO

CREATE CERTIFICATE DatabaseName\_Cert

WITH SUBJECT = 'SubjectHere',

EXPIRY\_DATE = '20200315';

GO

1d – Backup New cert

Use Master

go

BACKUP CERTIFICATE DatabaseName\_Cert TO FILE = 'd:\TDE\CertName.cer'

WITH PRIVATE KEY ( FILE = 'd:\TDE\KeyName\_Key.pvk' ,

ENCRYPTION BY PASSWORD = 'StrongPWHere' );

GO

1e – Alter Database Encryption & Drop current Key

ALTER DATABASE Db1

SET ENCRYPTION OFF;

GO

/\* Wait for decryption operation to complete, look for a

value of 1 in the query below. \*/

SELECT encryption\_state

FROM sys.dm\_database\_encryption\_keys;

GO

select \* from sys.databases

USE Db1;

GO

DROP DATABASE ENCRYPTION KEY;

GO

1f – Associate Database with new cert & set encryption

USE Db1;

GO

CREATE DATABASE ENCRYPTION KEY

WITH ALGORITHM = AES\_128

ENCRYPTION BY SERVER CERTIFICATE DatabaseName\_Cert

GO

--Set Encryption ON

Alter Database Db1

Set ENCRYPTION ON;

Go

1g – Backup Database

Use your preferred method to back up the database.

Ex. Backup Database DB1 to Disk = 'D:\TDE\Db1\_03152016.bak'

1h – Move Cert, Keys, Backup to Destination

Move the certificate, key, and backup to the destination server if needed.

1i – Alter Database Encryption & Drop current Key

ALTER DATABASE Db1

SET ENCRYPTION OFF;

GO

/\* Wait for decryption operation to complete, look for a

value of 1 in the query below. \*/

SELECT encryption\_state

FROM sys.dm\_database\_encryption\_keys;

GO

USE Db1;

GO

DROP DATABASE ENCRYPTION KEY;

GO

1j - Associate Database with old cert & set encryption

CREATE DATABASE ENCRYPTION KEY

WITH ALGORITHM = AES\_128

ENCRYPTION BY SERVER CERTIFICATE OldCert;

GO

--Set Encryption ON

Alter Database Db1

Set ENCRYPTION ON;

Go

1k – Review encryption status

SELECT encryption\_state, \*

FROM sys.dm\_database\_encryption\_keys;

GO

select \* from sys.database

**Begin Destination Process**

**2 – Destination**

2a – Determine if there is already a Master Key

These scripts will help you determine what keys and certificates are on the database. Be sure to run this in the Master as well as in the Source Database so you can see the Master Key as well. Note that an encryption state of 3 is encrypted.

use master

go

select \* from sys.symmetric\_keys

select \* FROM sys.dm\_database\_encryption\_keys

2b – Create Master if one doesn’t exist

USE master

GO

CREATE MASTER KEY

ENCRYPTION BY PASSWORD = 'StrongPasswordHere'

GO

2c – Create Cert/Key from Source files

Use master

go

CREATE CERTIFICATE NewDatabase\_Cert

FROM FILE = 'd:\temp\TDE\_TEST\CertName.cer'

WITH PRIVATE KEY (FILE = 'd:\temp\TDE\_TEST\KeyName\_Key.pvk',

DECRYPTION BY PASSWORD = 'StrongPWHere')

\*\*Note that the Decryption password here is the one we used to encrypt the backup from earlier.

3d – Restore Database

Restore the database with your preferred method.

3 e – Review encryption status

Select encryption\_state, \*

FROM sys.dm\_database\_encryption\_keys;

GO

select \* from sys.databases