




# SQL Server Encryption

Ben Miller

ben@benmiller.net

Twitter: @DBAduck

Blog: http://dbaduck.com

http://PowerShellDBA.com

SQL *intersection*

# Introduction

- Using SQL Server since version 4.2
- SQL Server MCM
- SQL Server MVP
- Been in IT for over 25 years
- Automation and Integration are specialties
- MCAD: C# and Web Development
- MCSE: Data Platform
- Independent Consultant – ben@benmiller.net

SQL intersection

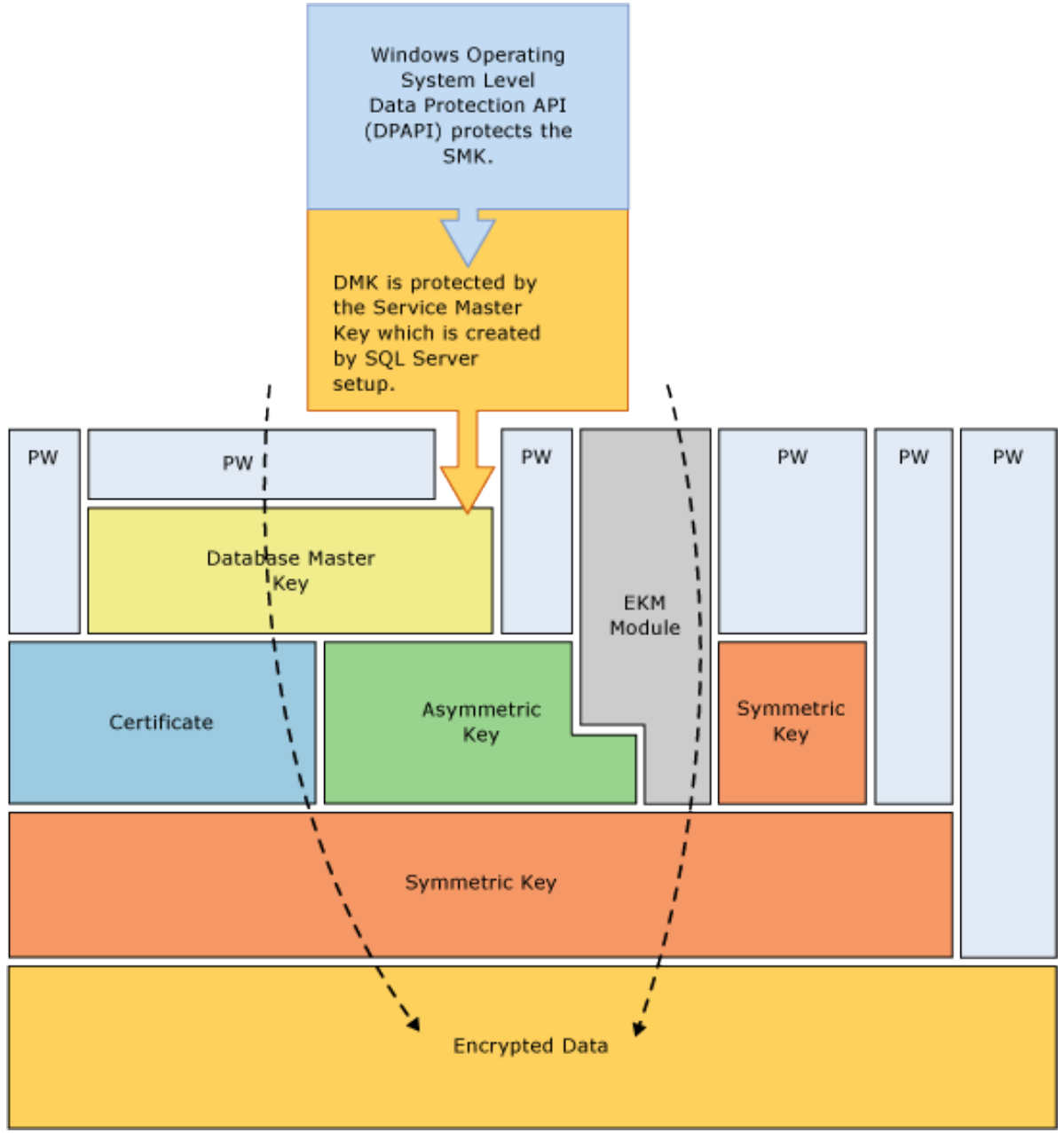
# Agenda

- **Encryption Hierarchy**
- **Keys**
  - Symmetric Keys
  - Asymmetric Keys
  - Database Encryption Key
- **Certificates**
- **TDE**
- **Encrypted Backup**
- **Cell Encryption**

SQL intersection

# Encryption Hierarchy


SQL intersection




SQL intersection

# Symmetric Encryption Keys

- **Symmetric Keys**
  - Do not use RC4 and RC4_128 algorithms
  - Symmetric Keys created with a Password are created with TRIPLE_DES encryption
  - Symmetric keys created with ALGORITHM = TRIPLE_DES_3KEY use TRIPLE DES with a 192-bit key
  - Symmetric keys created with ALGORITHM = TRIPLE_DES use TRIPLE DES with a 128-bit key
- **Permissions**
  - ALTER ANY SYMMETRIC KEY permission on the database.
  - If AUTHORIZATION is specified, requires IMPERSONATE permission on the database user or ALTER permission on the application role.
  - If encryption is by certificate or asymmetric key,
    - Requires VIEW DEFINITION permission on the certificate or asymmetric key.
    - Only Windows logins, SQL Server logins, and application roles can own symmetric keys. Groups and roles cannot own symmetric keys.

SQL intersection

# Asymmetric Encryption Keys

- **Assymmetric Keys**
  - Only Windows logins, SQL Server logins, and application roles can own asymmetric keys. Groups and roles cannot own asymmetric keys
  - This is a Key Pair and Encrypted by Master Key or Password or EKM Provider
  - Can be RSA_512, RSA_1024, RSA_2048
- **Permissions**
  - Requires CREATE ASYMMETRIC KEY permission on the database.
  - If the AUTHORIZATION clause is specified, requires IMPERSONATE permission on the database principal, or ALTER permission on the application role.
  - Only Windows logins, SQL Server logins, and application roles can own asymmetric keys. Groups and roles cannot own asymmetric keys.

SQL intersection

# Encryption Keys (2)

- **Database Encryption Key**
  - Special Symmetric key used in TDE

- **Notes:**
- **You should SALT the values on encryption so that comparisons or replacement hacks cannot take place.**
- **For best performance, encrypt data using symmetric keys instead of certificates or asymmetric keys**
- **Database master keys are protected by the Service Master Key**
- **An Extensible Key Management (EKM) module holds symmetric or asymmetric keys outside of SQL Server**
- **The Service Master Key and all Database Master Keys are symmetric keys**

SQL intersection

# Certificates

- Public Key / Private Key
- Only requirement is a Subject and Name
- Expiry Date is 1 year from creation unless you specify
- Expiry Date is ignored by TDE
- Can be used by encryption and the only option for Cell if you use a Function to encrypt/decrypt
- Backups are VERY IMPORTANT (public and private key)

SQL intersection

# What is TDE?

- **Transparent Data Encryption**
- **Encrypted Database "At Rest"**
- **Encryption with AES or 3DES algorithms**
  - AES_128, AES_192, AES_256, TRIPLE_DES_3KEY
- **Encryption is performed at the Page Level**
  - Data and Log files
- **When one database is encrypted, tempdb is encrypted by default.**
- **FILESTREAM data is not encrypted when TDE is enabled**

SQL intersection

# Requirements for TDE - Review

- **SQL Server 2008/2012/2014 Enterprise Edition**
- **Master database – Master Key**
- **Master database – Certificate**
- **User database – Database Encryption Key**
  - Stored in the boot record of database for recovery
- **ALTER DATABASE SET ENCRYPTION ON**

SQL intersection

# Benefits of TDE

- No schema changes like cell level encryption
- Page Level encryption
- MSFT estimates degradation at 3-5% instead of 20-28% with cell level encryption **
- Secure backups by default
- Invisible to the User
- “At Rest” Encryption

SQL intersection

# Disadvantages of TDE

- Backup Compression no longer effective
- Enterprise Edition required
- With Cell Level encryption you have finer control over encrypted elements
- With one database encrypted, TempDB is encrypted for ALL databases
- Even when all DBs are decrypted, Instance Restart required to remove encryption from TempDB

SQL intersection

# Backup Encryption

- **Requirements**
  - Certificate or Asymmetric Key in EKM Provider
  - New Media Set
- **Backup Compression still effective**

SQL intersection

# Show me the money!

***Demo***

SQL intersection

# System DMVs for Encryption

- sys.key_encryptions
- sys.symmetric_keys
- sys.certificates
- sys.dm_database_encryption_keys
- sys.asymmetric_keys

SQL intersection

# Things to watch for…

- BACKUP database before enabling TDE
- You are allowed to drop a certificate in the hierarchy used for TDE **
- BACKUP all Certificates and Private Keys
- BACKUP all Keys for safe keeping
- In Mirroring and Replication, both databases are encrypted

- ** Prior to SQL 2008 R2 SP2 you can drop the certificate because there is no dependency held.

SQL intersection

# Ticking Time Bombs

- **Certificates can be dropped (pre SQL 2008R2 SP2) even if TDE is enabled on a database.**
- **Database will still function when certificate is dropped until restart. Ensure you have a backup.**

SQL intersection

# Summary

- Only available in Enterprise Edition
- TDE is enabled via Keys and Certificates
- Data is encrypted “At Rest” not over the wire
- Backups of Encrypted database are encrypted
- Protect your Encryption assets (Keys, Certificates, etc)

SQL intersection

# Resources

- **SQL PASS (http://www.sqlpass.org)**
  - SQL Saturday in your area (sqlsaturday.com)
  - SQL Server User Groups in your area
- **Microsoft SQL Server website**
  - http://www.microsoft.com/sql
- **SSWUG.org**
  - http://www.sswug.org
  - https://msdn.microsoft.com/en-us/library/bb934049.aspx

SQL intersection





## Contact Information

Ben Miller

ben@benmiller.net

Twitter: @DBAduck

Blog: http://dbaduck.com

http://PowerShellDBA.com

SQL *intersection*