Security Talk:
Protecting Your Data Using Encryption in SQL Server

Il-Sung Lee
Senior Program Manager
Microsoft Corporation
ilsung@microsoft.com
What We Will Cover

- Cell-Level Encryption
- Transparent Data Encryption
- Extensible Key Management (EKM)
- Channel Encryption
Data Encryption

• Why consider encryption?
  – Additional layer of security
  – Required by some regulatory compliance laws

• In Microsoft® SQL Server® 2000
  – Channel Encryption only

• Since Microsoft® SQL Server® 2005
  – Built-in support for data encryption
  – Support for key management

• Encryption additions in Microsoft® SQL Server®
  – Transparent Data Encryption
  – Extensible Key Management
Cell-Level Encryption
Cell-Level Encryption

- Encryption/Decryption built-ins
- Symmetric and asymmetric keys, certificates
  - Creation data definition language (DDL)
  - Symmetric keys and private keys stored encrypted
- Keys secured by:
  - User passwords
  - Automatically (SQL Server key management)
- Choice of algorithms
  - DES, TRIPLE_DES, RC2, RC4, RC4_128, AES (128, 192, or 256)
Key Hierarchy

Operating System Level
Data Protection API (DPAPI)

DPAPI encrypts Service Master Key

Service Master Key encrypts Database Master Key

Password

SQL Server 2005 Instance Level
Service Master Key

SQL Server 2005 Database Level
Database Master Key

Password

Certificates

Asymmetric Keys

Password

Symmetric Keys

Password

Table

Col1

Col2

Value
demonstration

Clinic Encryption
Don’t Forget Module Signing

• Need ALTER ANY LOGIN server permission to ALTER LOGIN
• Need to GRANT ALTER ANY LOGIN TO Alice? – No!

ALTER LOGIN Bob ENABLE

Alice (non-privileged login)
Don’t Forget Module Signing

- Alice has permission to call SP
- SP run under Alice’s context but with elevated privilege
- SP protected against tampering

Alice (non-privileged login)
Best Practices

- Make signing modules a habit
- Do not encrypt all of your data
- Use symmetric encryption
- Plan carefully
  - Key management is very important
  - Understand changes to existing code needed
  - Consider key size and algorithm on CPU
Transparent Data Encryption
Transparent Data Encryption (TDE)

- Encryption/decryption at database level
- Database encryption key (DEK) is encrypted with:
  - Service Master Key
  - Key residing in a Hardware Security Module (HSM)
- DEK must be decrypted to attach database files or restore a backup
TDE – Key Hierarchy

- **Database Master Key** encrypts Certificate in Master Database
- **Service Master Key** encrypts **Database Master Key**
- **Database Encryption Key** is encrypted by **Service Master Key**
- **Certificate** encrypts **Database Encryption Key**
- **Password** is used to encrypt **Certificate**
- **Data Protection API (DPAPI)** encrypts **Service Master Key**
- **Operating System Level** provides security for **Service Master Key**

Diagram:
- **SQL Server 2008 Instance Level**
  - **Service Master Key**
  - **Database Master Key**
- **SQL Server 2008 Master Database**
  - **Certificate**
  - **Database Encryption Key**
- **SQL Server 2008 User Database**
Reasons to Use TDE

- Protects data at rest
  - Data files, log files, backup
- Entire database is protected
- No application changes!
  - No restrictions with indexes or data types (except Filestream)
- Performance cost is small
- Storage space size unchanged
TDE Considerations

• Compatible with Database Compression
• Not recommended with Backup Compression
• Database Mirroring
  – Copy certificate from primary to mirror
• Log files are not retroactively encrypted
  – Encryption begins at next virtual log file (VLF) boundary
• Tempdb is encrypted when 1 db in instance uses TDE
• Enterprise only
demonstration

Enabling Transparent Data Encryption
Extensible Key Management
Extensible Key Management

- Key storage, management, and encryption done by HSM module
- EKM key appears same as normal key to app
- SQL EKM Provider dynamic-link library (DLL) provided by HSM vendor
Advantages of Using EKM

• Security
  – Data and keys are physically separated
  – Centralized key management for enterprise
  – Separation of duties between database owner and data owner

• Performance
  – Pluggable hardware encryption boards
Cryptographic Providers and Keys

- EKM providers are server objects
- EKM keys are very similar to native keys
  - Managed using the same TSQL
  - Visible in the same catalogs
  - Data encryption with standard built-ins
  - Used to encrypt SQL native keys
EKM Key Hierarchy in SQL Server 2008

- **HSM**
  - Symmetric key
  - Asymmetric key

- **EKM Symmetric key**
  - EKM Asymmetric key

- **SQL Server**
  - Data
  - Data
  - Native Symmetric key
  - TDE DEK key
EKM Vendor Support

- And others coming soon…

- SafeNet
- www.arx.com
- Thales
- NCipher
Channel Encryption
Channel Encryption

- Full Secure-Sockets Layer (SSL) encryption
- Login credentials always encrypted
- Self-signed certificates
Protecting Your Data

- Encrypt data to provide additional layer of security
- Use TDE to protect data at rest
- Protect your keys!
- Leverage EKM to consolidate key management and separate data from keys
- Protect data in transit
For More Information...

- Visit TechNet at [www.microsoft.com/technet](http://www.microsoft.com/technet)
- For additional information on books, courses, and other community resources that support this session visit
Questions and Answers

• Submit text questions using the “Ask” button
• Don’t forget to fill out the survey
• For upcoming and previously live webcasts: www.microsoft.com/webcast
• Got webcast content ideas? Contact us at: http://go.microsoft.com/fwlink/?LinkId=41781
Your potential. Our passion.