Data Encryption in SQL 2005

M. Choirul Amri, MCT, MCITP, MVP
Senior Consultant Trainer
Avantus Training

choirul@avantustraining.com
http://choirulamri.or.id

Credit to…

Lara Rubbelke
http://blogs.digineer.com/blogs/larar/default.aspx

SQL Server Books Online

What We Will Cover

- How to Encrypt SQL Server object structure
- How SQL Server 2005 data implements encryption
- How to Secure SQL Server Endpoint

Agenda

- Available Encryption Options
- Encrypt SQL Server Object
- SQL Server 2005 Encryption 101
- Encryption Architecture
- Securing Endpoint

Available Encryption Options

- Encrypt SQL Server Object
  - Encrypt object definition
  - Applicable to view and stored procedure
  - Available in SQL Server 2000 and 2005
- Encrypt SQL Server Data
  - Encrypt data on column level
  - Use Symmetric or Asymmetric key
  - Only Available in SQL Server 2005
  - Write your own SP or use 3rd party for SQL 2000

Encrypt SQL Server Object

- Hide the sensitive business logic on view and stored procedure
- This is one way encryption, no decryption method available
- Make sure to backup the object script before encrypt

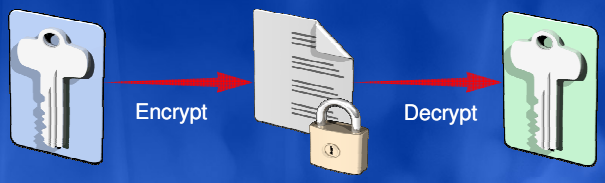
```
CREATE PROCEDURE HumanResources.usp_YearEndBonus
WITH ENCRYPTION
AS…..

ALTER VIEW HumanResources.vw_SpecialDiscount
WITH ENCRYPTION
AS…..
```

Encrypt SQL Server Data

- SQL Server encryption 101
  - Understanding Key and Certificate
  - Key Hierarchy
- Data encryption method
  - Hashing
  - Asymmetric encryption
  - Symmetric encryption

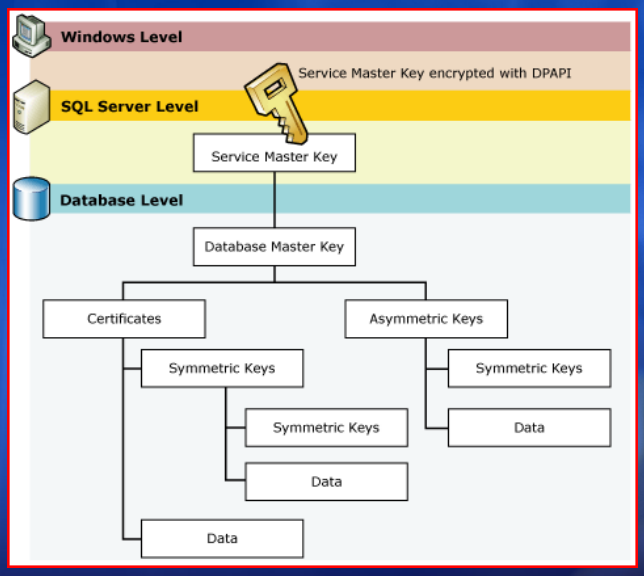
SQL Server 2005 Encryption 101

- Key

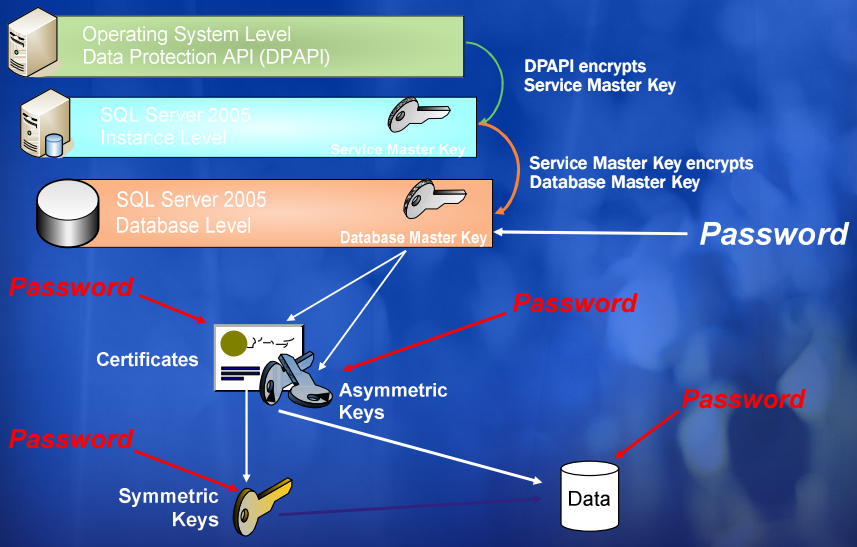


- Certificates
  - Associates a public key with entity that holds that key
  - Identity, validity periods, digital signature

SQL Server 2005 Key Hierarchy



Encryption Architecture



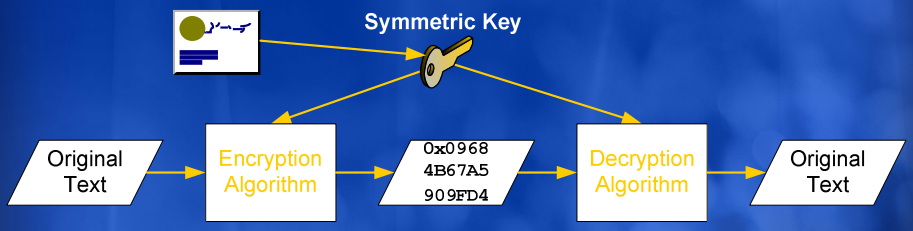
Data Encryption Method 1

- HashBytes()

- One-way hashing
- Deterministic
- Good when you want the identifying aspect of the data but not the actual value
  - Passwords
  - Creating Salts

Data Encryption Method 2

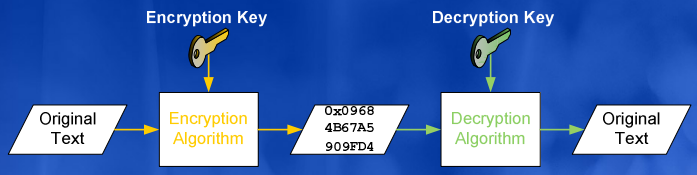
- SQL Server 2005 Symmetric Encryption



- Use the same key for encrypt and decrypt
- SQL Server can secure symmetric keys with a password, symmetric key, certificate, and/or asymmetric key
- Nondeterministic
- Faster than asymmetric encryption (can be more than 1,000 times faster!)
- Supports encryption algorithms: DES, TRIPLE_DES, RC2, RC4, RC4_128, DESX, AES (128, 192, or 256)

Data Encryption Method 3

- SQL Server 2005 Asymmetric Encryption



- Use different key for encrypt and decrypt
- More secure encryption
- Nondeterministic
- Useful to protect symmetric keys or create digital signatures
- Certificates and asymmetric keys both provide the same asymmetric encryption capabilities
- RSA encryption algorithm can be used for key sizes: 512-bit, 1024-bit, 2048-bit

Encrypting Your Data

- Create master key with strong password if not yet exists

```
CREATE MASTER KEY ENCRYPTION BY
PASSWORD = `jdsghe%6'
```

- Create certificate
- Create encryption key: symmetric or asymmetric
- Encrypt the data with EncyptByKey function
- To read encrypted data: use DecryptByKey function

Creating Certificate and Key

- Create certificate

```
CREATE CERTIFICATE HumanResources037
WITH SUBJECT = 'my cert key'
```

- Create encryption key: symmetric or asymmetric

```
CREATE SYMMETRIC KEY SSN_Key_01
    WITH ALGORITHM = AES_256
    ENCRYPTION BY CERTIFICATE HumanResources037;
```

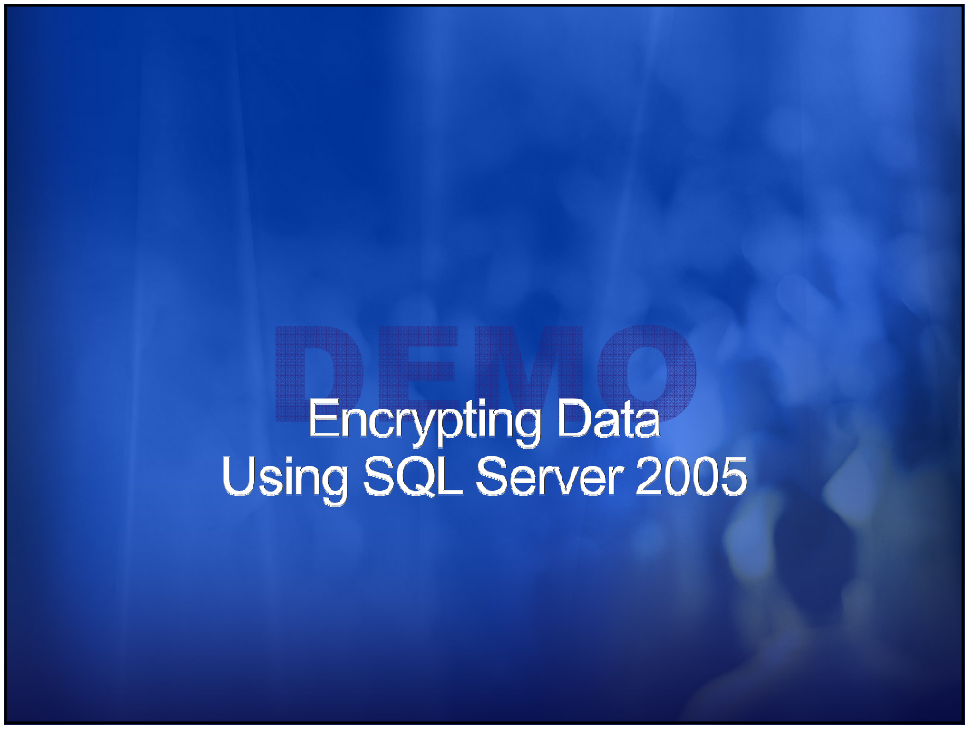
Encrypt and Decrypt The Data

- Encrypt the data

```
UPDATE HumanResources.Employee
SET EncryptedNationalIDNumber =
   EncryptByKey(Key_GUID('SSN_Key_01'),
   NationalIDNumber);
```

- Reading encrypted data

```
SELECT CONVERT(nvarchar,
   DecryptByKey(EncryptedNationalIDNumber))
    AS "Decrypted ID Number"
    FROM HumanResources.Employee;
```



Best Practices Secure Encryption Implementation

- Do not encrypt all of your data
- Test hardware with encryption algorithms—larger keys may stress CPU more than smaller keys
- Use symmetric encryption
  - Encrypt symmetric keys using asymmetric keys
  - Use the strongest encryption algorithm you can based on performance requirements
    - AES is the symmetric algorithm of choice
    - Do not use anything lower than 128-bit symmetric keys

Additional Best Practices

- Secure your server and instance
- Secure your private keys
  - Grant least privileges necessary
- Back up your Master Keys!!!
  - Store in a secure, off-site location
- Ensure user database backups are successful and a reliable backup exists

Securing Endpoint

- SQL Server 2005 exposes the services with endpoint
- Endpoint support HTTP and TCP protocol for the following purposes:
  - TSQL
  - Database Mirroring
  - Service Broker
  - SOAP
- Limit access to endpoint wherever possible, specially for public endpoint such as HTTP for SOAP or service brocker

Securing Endpoint

- Use DENY statement for specific endpoint and login

```
USE master;
DENY CONNECT ON ENDPOINT::mirroringendpoint
to john
```

Session Summary

- Microsoft SQL Server 2005 provides many methods to encrypt your sensitive data
- Carefully plan which columns to encrypt
- Use symmetric encryption
  - Encrypt symmetric keys using asymmetric keys/ certificates
- Carefully plan data access paths and design a solution which ensures performance and scalability

Additional Encryption Resources

Expert SQL Server 2005 Development
by Adam Machanic, Hugo Kornelius, and Lara Rubbelke

TechNet Webcast: Encryption and Key Management Using SQL Server 2005
http://msevents.microsoft.com/cui/WebCastEventDetails.aspx?EventID=1032293593&EventCategory=4&culture=en-US&CountryCode=US

Protect Sensitive Data Using Encryption in SQL Server 2005
http://www.microsoft.com/technet/itsolutions/msit/security/sqldatsec.mspx

Lara Rubbelke's blog
http://blogs.digineer.com/blogs/larar/default.aspx

Laurentiu Cristofor's blog
http://blogs.msdn.com/icris/

Raul Garcia's blog
http://blogs.msdn.com/raulga/default.aspx


Q&A