

SQL Server 2005 Books Online (September 2007)

## Encryption Hierarchy

 [Send Feedback](#)

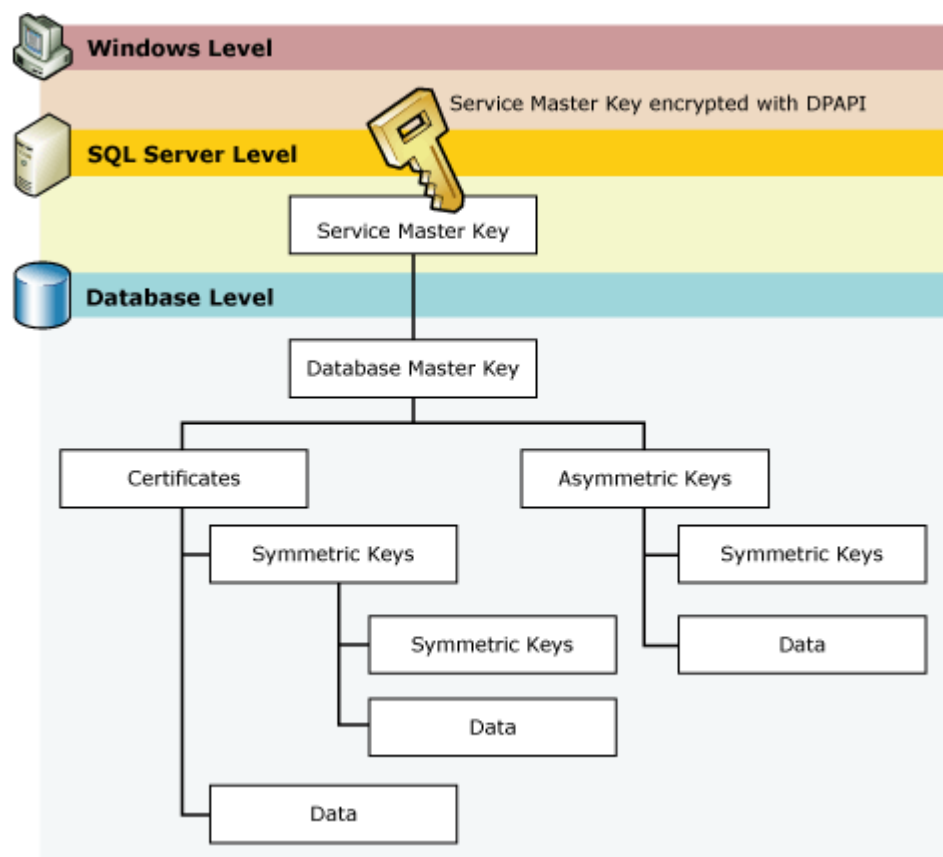
 [See Also](#)

 Collapse All

[Security Considerations for Databases and Database Applications](#) >

SQL Server 2005 encrypts data with a hierarchical encryption and key management infrastructure. Each layer encrypts the layer below it by using a combination of certificates, asymmetric keys, and symmetric keys. As shown in the following illustration, the encryption hierarchy is parallel to the hierarchy of securable objects that are described in [Permissions Hierarchy](#).

The following illustration shows that each layer of the encryption hierarchy encrypts the layer beneath it. The top layer, the Service Master Key, is encrypted with the Windows DP API.



## Encryption Mechanisms

SQL Server 2005 provides the following mechanisms for encryption:

- Certificates
- Asymmetric keys
- Symmetric keys

### Certificates

A public key certificate, usually just called a certificate, is a digitally-signed statement that binds the

value of a public key to the identity of the person, device, or service that holds the corresponding private key. Certificates are issued and signed by a certification authority (CA). The entity that receives a certificate from a CA is the subject of that certificate. Typically, certificates contain the following information.

- The public key of the subject.
- The identifier information of the subject, such as the name and e-mail address.
- The validity period. This is the length of time that the certificate is considered valid.

A certificate is valid only for the period of time specified within it; every certificate contains **Valid From** and **Valid To** dates. These dates set the boundaries of the validity period. When the validity period for a certificate has passed, a new certificate must be requested by the subject of the now-expired certificate.

- Issuer identifier information.
- The digital signature of the issuer.

This signature attests to the validity of the binding between the public key and the identifier information of the subject. (The process of digitally signing information entails transforming the information, as well as some secret information held by the sender, into a tag called a signature.)

A primary benefit of certificates is that they relieve hosts of the need to maintain a set of passwords for individual subjects. Instead, the host merely establishes trust in a certificate issuer, which may then sign an unlimited number of certificates.

When a host, such as a secure Web server, designates an issuer as a trusted root authority, the host implicitly trusts the policies that the issuer has used to establish the bindings of certificates it issues. In effect, the host trusts that the issuer has verified the identity of the certificate subject. A host designates an issuer as a trusted root authority by putting the self-signed certificate of the issuer, which contains the public key of the issuer, into the trusted root certification authority certificate store of the host computer. Intermediate or subordinate certification authorities are trusted only if they have a valid certification path from a trusted root certification authority.

The issuer can revoke a certificate before it expires. Revocation cancels the binding of a public key to an identity that is asserted in the certificate. Each issuer maintains a certificate revocation list that can be used by programs when they are checking the validity of any given certificate.

The self-signed certificates created by SQL Server follow the X.509 standard and support the X.509 v1 fields.

## Asymmetric Keys

An asymmetric key is made up of a private key and the corresponding public key. Each key can decrypt data encrypted by the other. Asymmetric encryption and decryption are relatively resource-intensive, but they provide a higher level of security than symmetric encryption. An asymmetric key can be used to encrypt a symmetric key for storage in a database.

## Symmetric Keys

A symmetric key is one key that is used for both encryption and decryption. Encryption and decryption by using a symmetric key is fast, and suitable for routine use with sensitive data in the database.

## See Also

### Concepts

[Permissions Hierarchy](#)  
[Securables](#)

### Other Resources






[Security Functions \(Transact-SQL\)](#)  
[Encryption How-to Topics](#)

### Help and Information

[Getting SQL Server 2005 Assistance](#)

#### Documentation Feedback

Microsoft values your feedback. To rate this topic and send feedback about this topic to the documentation team, click a rating, and then click **Send Feedback**. For assistance with support issues, refer to the technical support information included with the product.

Poor	1	2	3	4	5	Outstanding
						

To e-mail your feedback to Microsoft, click here:

[Send Feedback](#)

[© 2007 Microsoft Corporation. All rights reserved.](#)