

ITPro<sup>TM</sup>  
SERIES

Windows IT Pro

 **eBooks**

Keeping Your Business  
SAFE from Attack:

# Encryption and Certificate Services

By Jan De Clercq

**Microsoft<sup>®</sup>**



**Microsoft®**

## Contents

|  |           |
|--|-----------|
| <b>Chapter 2 Windows PKI Components</b> .....              | <b>19</b> |
| <b>Certificate Server</b> .....                            | <b>19</b> |
| Certificate Server Architecture .....                      | 19        |
| CA Installation Modes .....                                | 22        |
| <b>Registration Authorities</b> .....                      | <b>24</b> |
| <b>Active Directory</b> .....                              | <b>25</b> |
| <b>CryptoAPI and Cryptographic Service Providers</b> ..... | <b>27</b> |
| CryptoAPI Architecture .....                               | 27        |
| Cryptographic Service Providers .....                      | 29        |
| <b>Certificate Templates</b> .....                         | <b>30</b> |
| <b>Certificate Storage</b> .....                           | <b>32</b> |
| <b>Private Key Storage</b> .....                           | <b>35</b> |
| Software-Based Storage .....                               | 36        |
| Hardware-Based Storage .....                               | 36        |

## Chapter 2:

# Windows PKI Components

This chapter introduces the core components of a Windows Public Key Infrastructure (PKI): the Certificate Server, registration authorities, Active Directory (AD), and the CryptoAPI. It also explains certificate templates and Windows PKI certificate and private-key storage.

## Certificate Server

The Windows Server 2003 Certificate Server provides the following core services:

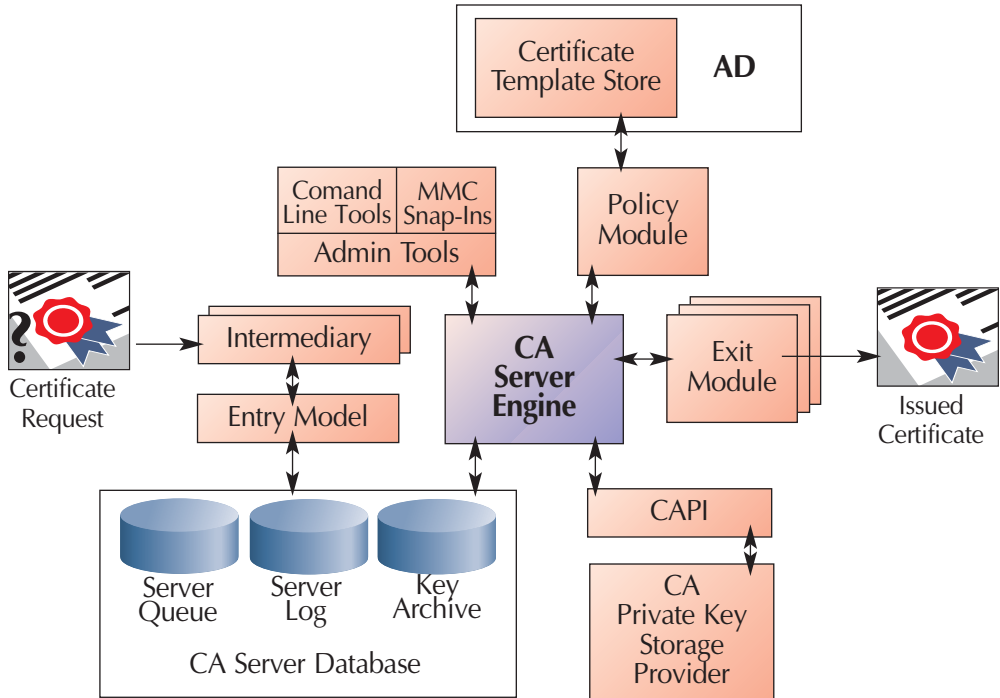
- Receives and processes certificate requests.
- Identifies and validates certificate requests.
- Issues certificates according to the PKI configuration settings.
- Renews and revokes certificates.
- Publishes certificates.
- Creates and publishes CRLs.
- Logs all certificate and CRL transactions into a log database.

A brand new function of the Windows Server 2003 Certificate Authority (CA) is its capability to take care of secure private-key archival and recovery.

## *Certificate Server Architecture*

The architecture of the Microsoft Certificate Server is illustrated in Figure 2-1. The design is largely identical to that used in the Windows 2000 edition of the Microsoft Certificate Server. A new feature not available in previous versions is a modified CA database layout that lets the CA cope with private-key archival and recovery.

**Figure 2-1:**  
*Certificate server architecture*



At the heart of the Windows Certificate Server sits an engine (`certsrv.exe`) that generates certificates and certificate revocation lists (CRLs), and directs the message flow among the other components. The engine communicates with three important modules: the entry module, the policy module, and the exit module.

- The entry module accepts PKCS7-, PKCS10-, and CMC-formatted certificate requests. This module's only job is to place these requests in a queue for treatment by the policy module.
- The policy module implements and enforces the CA policy rules as set forward by the CA administrator. This module informs the CA engine about the layout of a certificate and decides whether a certificate request should be issued, denied, or left pending. To make the latter decision and to retrieve certificate layout information, the policy module can call upon information stored in a directory or database. Windows Server 2003 comes with a policy module (called `certpdef.dll`) that supports two policy types: enterprise and standalone. We discuss details about these two policy types and how to configure them later in this chapter. To check out the policy module your CA is using, look at the CA object properties (Policy Module tab) in the Certification Authority Microsoft Management Console snap-in.

- The exit module distributes and publishes certificates, certificate chains, CRLs and delta CRLs. The exit module can write public-key infrastructure (PKI) data to the file system or transport it across HTTP, Lightweight Directory Access Protocol (LDAP), or remote procedure call (RPC) to a remote location. The Windows Server 2003 CA can support multiple exit modules. This capability enables the CA to publish and distribute certificates, certificate chains, CRLs, and delta CRLs to different locations in parallel: LDAP directories, file shares, Web directories, or even ODBC-compliant databases. The default Windows Server 2003 CA exit module is called `certxds.dll`. To check the exit modules installed on your CA, look at the CA's (Exit Module tab) in the Certification Authority Microsoft Management Console snap-in.

The exit module and the policy module are both customizable and replaceable. If either module doesn't correspond to the needs of an organization, the business can develop its own modules in C++ or Visual Basic (VB) and plug the customized modules into the CA architecture. The Windows Server 2003 platform software development kit (SDK) documents these options.

You can configure the policy and exit modules using either the Certification Authority Microsoft Management Console snap-in or the `certutil` command-line utility. Using the properties of the CA object in the Certification Authority Microsoft Management Console snap-in, you can do things such as add another exit module, configure X.509 certificate extensions—CRL and delta CRL distribution points (CDP) and Authority Information Access (AIA) points—and Configure CRL and delta CRL publication parameters.

The Certificate Server uses a database to store certificate transactions and status information, certificates, and optionally archived private keys. The database (`<CAName>.edb`) is, by default, located in the `system32\certlog` folder. The Certificate Server engine communicates with its database through the `certdb.dll` file. In the Windows 2000 release of Certificate Server, Microsoft changed its database technology to JET Blue, the same technology used for AD and the Exchange databases. This switch gave the Windows 2000 CA a scalability injection.

Windows Server 2003 CA management is done primarily from the Certification Authority Microsoft Management Console snap-in. Another option is to use the `certutil.exe` command-line utility. Both administration tools communicate with the CA engine using the `certadm.dll` file.

The CA interacts with entities known as *intermediaries* to communicate with PKI clients. Intermediaries help PKI clients generate correctly formatted Public-Key Cryptography Standards (PKCS) #10 or Continuous Media Controller (CMC) certificate requests. An intermediary gathers user- and request-specific data that are required for a valid certificate request. For example, a request that is sent to a Windows Server 2003 enterprise CA should mention a certificate template. An intermediary can add a certificate template specification to the request. Intermediaries are bound to a specific transport protocol. Thanks to this constraint, the CA engine doesn't need to deal with different transport providers.

Examples of Windows Server 2003 intermediaries are the Web enrollment pages—an HTTP intermediary—and the Microsoft Management Console certificates snap-in, which calls on the Certificate Request Wizard—an RPC intermediary. The HTTP intermediary calls upon the `xenroll.dll` file—to generate private keys on the client machine—and the `scenroll.dll` file—to generate private

keys on a smart card. The RPC intermediary calls upon the `certcli.dll` file to perform these tasks. Examples of third-party remote automation (RA) software are discussed later on in this chapter.

For all cryptographic functions, including accessing and using the CA's private key, the CA calls upon the CryptoAPI. The CA's private key can be stored on hard disk or on a dedicated hardware device, such as a Hardware Security Module (HSM). The CryptoAPI is discussed later in this chapter.

### **CA Installation Modes**

When you install a Windows Server 2003 Certificate Server, you have the following installation options: You can install it as a root CA or a subordinate CA, and you can install it as an enterprise CA—AD integrated—or a standalone CA—non-AD integrated. Installing the Certificate Server in Enterprise mode provides full integration with AD. This configuration activates the Enterprise mode of the Windows Server 2003 CA policy module.

Let's have a look now at what Enterprise mode really means and how this setup differs from a CA installed in Standalone mode. Table 2-1 compares the default characteristics of a Windows Server 2003 standalone CA and an enterprise CA.

Enterprise mode typically targets enterprise PKI users who have an AD user account and authenticated to your AD infrastructure using the Kerberos protocol. These are also the environments in which you want to take advantage of an enterprise CA's capability to do automatic certificate enrollment for both users and machines. Conversely, standalone mode clearly targets external users (such as extranet users) who do not have an internal Windows account. Standalone CAs typically are used for offline CAs; enterprise CAs typically are used for issuing CAs. In the next chapter of this eBook, "Trust in Windows PKI," we'll explain why the mix of standalone and enterprise CAs in a PKI hierarchy is recommended.

Independent of the standalone/enterprise CA choice, I always recommend that you use Windows Server 2003 Enterprise Edition when you install an issuing CA and Windows Server 2003 Standard Edition when you install a root CA or an intermediate CA. Enterprise Edition includes features that are not available in the Standard Edition. These features include the issuing of certificates based on version 2 certificate templates (explained below), centralized private-key archival in the CA database, and role-separation support.

**Table 2-1: Windows Server 2003 Standalone CA versus Enterprise CA**

| <b>Windows Server 2003 Standalone CA</b>  | <b>Windows Server 2003 Enterprise CA</b>  |
|---|---|
| Non-AD integrated.  | AD integrated.  |
| Recommended for root CAs or intermediate CAs.   | Recommended for enterprise CAs.   |
| Extranet and Internet certificate-user oriented.  | Intranet certificate-user oriented.   |
| Can issue a limited set of certificate types and certificates requiring a custom Object Identifier (OID) in their extended key usage (EKU) extension; does not support certificate templates. | Can issue all Windows Server 2003 certificates defined in the Windows Server 2003 certificate templates Microsoft Management Console snap-in. Supports version 1 (Windows 2000 PKI) and version 2 (Windows Server 2003 PKI) certificate templates.  |
| User enrollment interface is Web-based. You can also use the certreq.exe command-line utility.  | User enrollment can be done using a Web interface or using the Microsoft Management Console certificates snap-in. Enrollment can also happen automatically using the certificate auto-enrollment feature. You can also use the certreq.exe command-line utility.  |
| Communication with the CA front end occurs across HTTP or HTTPS.  | Communication with the CA front end can use RPC/Distributed COM (DCOM) or HTTP/HTTPS.   |
| User has to enter identification information manually at certificate request time.  | User identification information is automatically retrieved from AD.   |
| Certificate enrollment approval happens automatically or manually. The CA has a single setting that controls this behavior for all certificate types.   | Certificate enrollment approval happens automatically or manually. This behavior can be controlled globally on the CA level or per certificate type using a certificate template setting. Also, the certificate approval process can use the AD authentication and access-control model through the ACLs that are set on certificate templates. |
| The certificate is downloaded to the user profile when it is manually retrieved from the CA Web site. By default, the CA does not publish certificates to AD.                                 | Depending on the certificate template, the certificate is automatically downloaded to the user profile and published to AD, or automatically published to AD.   |
| CRL and CA certificates can be published manually to AD.  | CRLs, Delta CRLs, CA, and cross-certification certificates are automatically published to AD.   |
| Does not automatically support AD-based certificate lookup and retrieval.   | Supports AD-based certificate lookup and retrieval.   |
| Can be installed on Windows Server 2003 Domain Controller, Member server, or standalone server (a server that's not a member of any domain).  | Can be installed on Windows Server 2003 Domain Controller or Member server.   |

## Registration Authorities

In large PKI setups, PKI users use a registration authority as the primary point of contact for a CA. A registration authority typically deals with PKI user identification and enrollment. Like its predecessors, Windows Server 2003 PKI does not include a true registration-authority function.

Windows Server 2003 supports limited RA functionalities through special “Enrollment agent” certificates (OID 1.3.6.1.4.1.311.20.2.1 – Certificate Request Agent). Enrollment agent certificates can be used for

- Smart-card bulk enrollment. Administrators with a special Certificate Request Agent certificate can enroll users’ certificates in bulk on smart cards and act as a registration authority for smart-card certificates.
- Integration with the Exchange Key Management Server (KMS). If your organization has implemented Exchange advanced mail security (Secure MIME—S/MIME) in combination with Microsoft Certificate Server, the Exchange KMS acts as a registration authority, identifying users and passing the certificate requests to the Windows 2000 CA.

The Web-enrollment interface that ships with the Windows Server 2003 PKI also can be considered a basic registration authority. You can install the Web-enrollment interface on a Web server, which can act as a kind of PKI registration or enrollment proxy, redirecting all certificate requests to a CA server.

Advanced registration authority support for Windows Server 2003 PKI is available from third-party software vendors. Table 2-2 below includes some examples. Table 2-2 does not provide an exhaustive list of registration-authority software; also, it is not the goal of this eBook to provide a feature comparison among the different registration-authority software products.

The reason Microsoft didn’t include a registration-authority function with the Windows Server 2003 OS probably has to do with the high level of customization that’s required to fit a registration authority to an organization’s needs. One organization may require that the registration authority is linked to its ERM system, another one may require smart-card enrollment and lifecycle management integration, and yet another one may require integration with its building access-control mechanism. Some regions also have special legal requirements for a registration-authority function. The European Union, for example, imposes strict rules for the implementation of a registration authority that is dealing with qualified certificates and certificates that are used for advanced digital signatures.

**Table 2-2: Registration Authority Software for Windows Server 2003 PKI**

| Company Name (Web site)   | Product Name | Key Features   |
|---|--------------|--|
| Alacris ( <a href="http://www.alacris.com">http://www.alacris.com</a> ) | idNexus      | <ul style="list-style-type: none"> <li>• Web-based registration authority administration interface</li> <li>• Smart-card enrollment support</li> <li>• Advanced reporting capabilities</li> </ul>  |
| Spyrus ( <a href="http://www.spyrus.com">http://www.spyrus.com</a> )    | SignalRA     | <ul style="list-style-type: none"> <li>• Microsoft Management Console-based registration authority administration interface</li> <li>• HSM support</li> <li>• Smart-card and security-token enrollment and management support</li> <li>• Advanced reporting capabilities</li> <li>• Dedicated audit log (SQL Server Database)</li> </ul> |

## Active Directory

A PKI uses a directory to store certificate revocation lists (CRLs); delta CRLs; and user, CA, and cross-certification certificates. When you set up a Windows Server 2003 PKI, AD is an obvious directory choice. AD is the only possible directory option if you want to take advantage of some typical Windows Server 2003 AD-based PKI features (e.g., the use of Microsoft enterprise CAs, certificate templates) or easily implement certain PKI-enabled applications (e.g., Windows smart-card logon). This does not mean that using another non-Microsoft directory to implement PKI-enabled applications, such as Windows smart-card logon, is impossible. Doing so may simply require more time and effort to set up and maintain the PKI-enabled application.

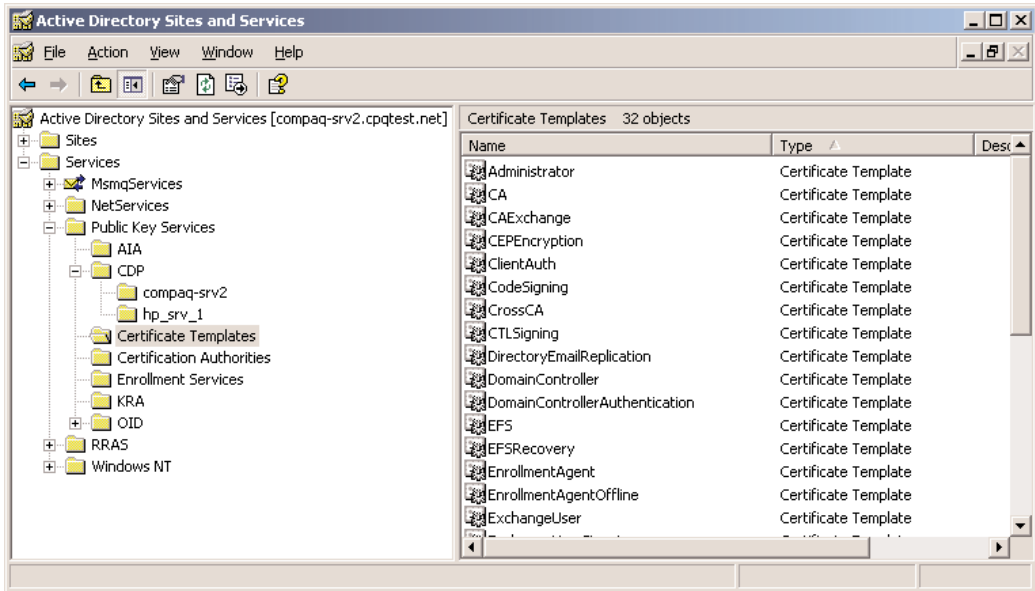
In general, Windows Server 2003 PKI is tightly integrated with AD. A nice example of this integration is the use of group policy objects to distribute trusted CA information to Windows 2000 or Windows XP Professional workstations. As explained in the previous sections, this integration is tight when one deploys Windows Server 2003 enterprise Certificate Servers. In that case, Windows Server 2003 PKI also uses AD to store CA- and PKI-related configuration information.

To look at the CA- and PKI-related entries in the AD configuration-naming context of your Windows Server 2003 forest, you can use several tools:

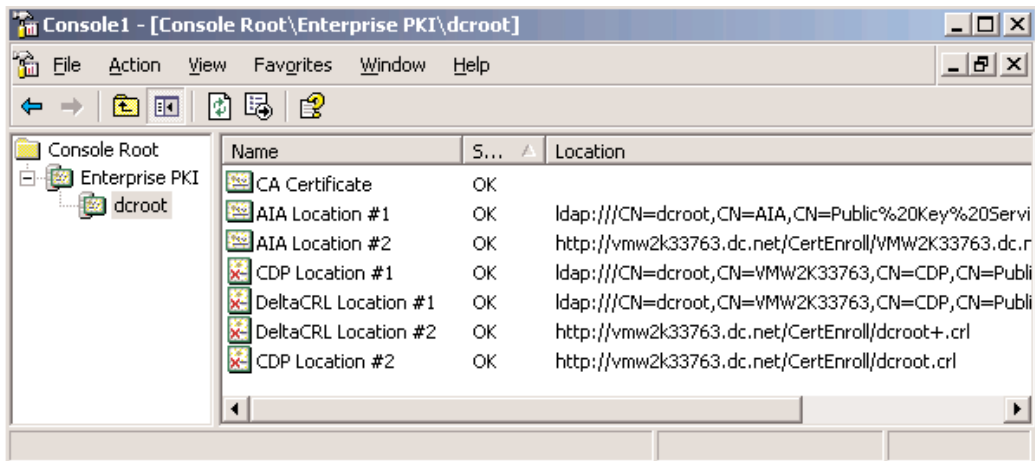
- The certutil command-line tool with the “-v -ds” switch.
- The Sites and Services Microsoft Management Console snap-in: All PKI-related, configuration-naming-context entries are visible from the Public Key Services container (as illustrated in Figure 2-2).
- The ADSIEdit tool (which comes with the Windows Server 2003 Support tools).
- The LDP tool (which comes with the Windows Server 2003 Support tools).
- The PKIView tool (which comes with the Windows Server 2003 Resource Kit): This tool can retrieve the content of the AIA, CDP, KRA (key recovery agent), Certification Authorities, Enrollment Services, and NTAAuthCertificates AD containers (as illustrated in Figure 2-3).

To query AD for user certificates, you can use the Search\Computers or people... function available from the Windows Server 2003 or Windows XP Start menu. The last tab of the user object properties is named Digital IDs and contains all the user certificates that are published in AD. You can use the same interface to export certificates to a file; afterward, you can import the certificate file into your personal certificate store.

**Figure 2-2:**  
*Querying AD for PKI-related information using the Sites and Services Microsoft Management Console snap-in*



**Figure 2-3:**  
*PKIView tool*



## CryptoAPI and Cryptographic Service Providers

The CryptoAPI is an API that comes bundled with the Windows Server 2003 and XP operating systems. CryptoAPI enables programmers to add cryptography-based security services, such as authentication, confidentiality, and integrity protection relatively easily to their applications. It also enables them to interact with certificates, private keys, and their secure storage providers. Most importantly, CryptoAPI hides the implementation details of the complex cryptographic algorithms from the programmer.

### *CryptoAPI Architecture*

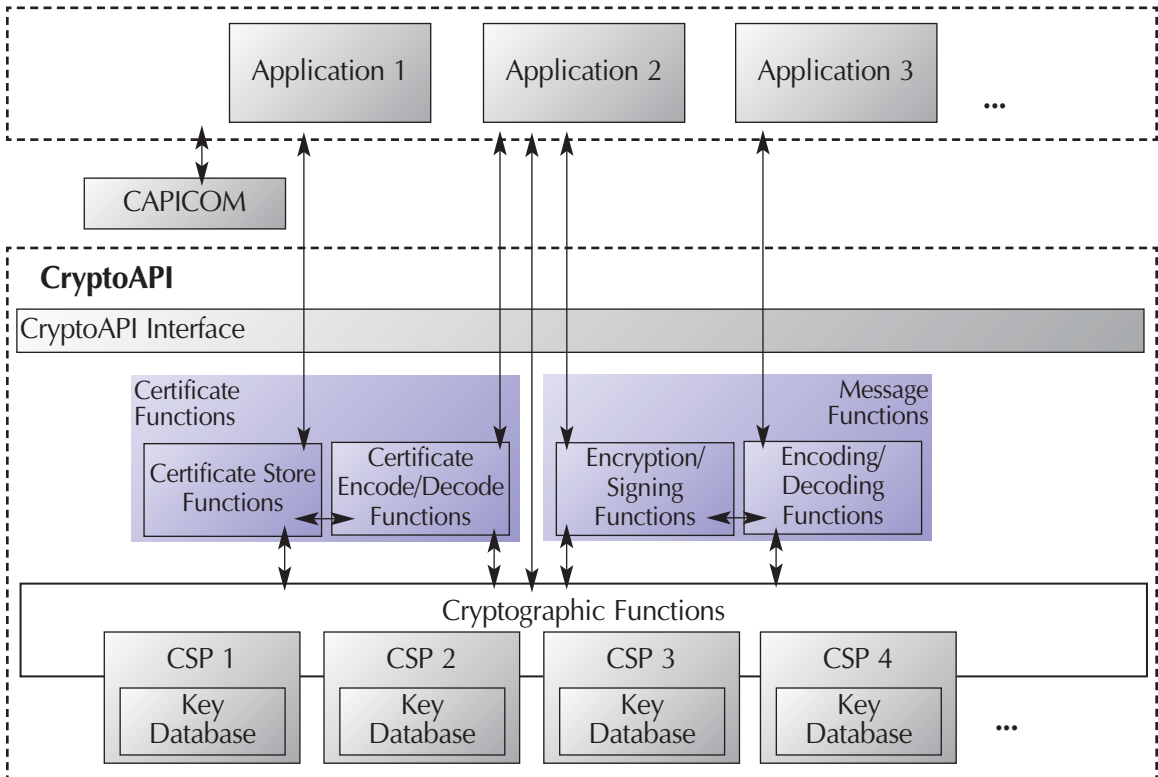
The CryptoAPI architecture (illustrated in Figure 2-4) consists of a programmatic interface, a set of software modules, and a set of pluggable Cryptographic Service Providers (CSPs). The CryptoAPI and its interfaces are fully documented in the Windows Server 2003 platform SDK.

The CryptoAPI software modules can be categorized by software modules that provide certificate-related functions and others that provide message-related functions:

- The modules that provide message-related functions can be called to generate cryptographic keys, perform hashing, digitally sign data, or perform data encryption and decryption. These modules also deal with message encoding and decoding services to and from the PKCS #7, PKCS #10, and Abstract Syntax Notation One (ASN.1) message formats.
- The modules that provide certificate-related functions deal with certificate management services. These modules can be called to generate, manage, and validate certificates, to interact with the certificate stores, and to encode and decode certificates.

The CryptoAPI functions can also be called through CAPICOM. CAPICOM is a COM client that can perform cryptographic functions using ActiveX and COM objects. CAPICOM can be used from applications created in Visual Basic, Visual Basic Scripting Edition, or C++. CAPICOM 2.0 includes support for the generation and verification of digital signatures, encryption and decryption of data, certificate store searching, hashing and the AES algorithm. CAPICOM is not available from a default Windows Server 2003 installation. You can download CAPICOM 2.1.0.1 (cc21inst.exe) from the Windows Platform SDK Redistributables Web site at <http://www.microsoft.com/downloads/details.aspx?FamilyID=860ee43a-a843-462f-abb5-ff88ea5896f6&DisplayLang=en>.

**Figure 2-4:**  
*CryptoAPI architecture*



CSPs are CryptoAPI software libraries that contain implementations of cryptographic algorithms and ciphers. The use of libraries creates a pluggable architecture; third-party vendors can plug their proper CSP into the OS and provide security services to applications.

CSPs can be implemented in both hardware and software. A hardware-based CSP implementation provides better security than a software-based implementation because hardware security devices such as smart cards, security tokens, and HSMs typically offer better protection against tampering.

To embed a CSP into Windows Server 2003 or XP, the CSP must be cryptographically signed by Microsoft. The CSP Development Kit available from Microsoft describes how to do this.

A CSP does more than just provide the implementation of a cryptographic cipher. CryptoAPI also deals with sensitive key (session key and private key) storage. It stores these keys into key databases that are embedded in the CSPs. The CSP key database contains a key container for each user; the container is named after the user's logon name. The key containers can be stored in the registry, on

the file system, or on a smart card. The key containers can never be accessed directly: They can be accessed only through the CryptoAPI and by using the appropriate CryptoAPI functions.

CSPs located on different machines cannot communicate directly. CryptoAPI allows sensitive keys to be exported from a CSP's key container and transported securely to another CSP. You can, for example, use a PKCS #12-formatted file to export and import certificates and private keys between different CSP key containers. When you export a private key to a PKCS #12 file, CryptoAPI forces you to use a password to protect the key. This password functions as a symmetric key and provides confidentiality protection for the exported private key.

### ***Cryptographic Service Providers***

Table 2-3 below gives an overview of the CSPs that come preinstalled with Windows Server 2003 and XP. To get an overview of all CSPs available on your machine, query the following registry folder: HKEY\_LOCAL\_MACHINE\Software\Microsoft\Cryptography\Defaults\Provider.

Windows Server 2003, XP, Windows 2000 Service Pack 2 (SP2), and later systems come preinstalled with the enhanced CSPs. The enhanced CSPs include support for 56-bit Data Encryption Standard (DES), 3 DES (112-bit 2key and 168-bit 3key), 16,384-bit RSA, 128-bit Release Candidate 2 (RC2), and RC4. The base CSPs support only 512-bit RSA, 40-bit RC2, and RC4. Before the release of Windows 2000 SP2, Windows 2000 users had to install the High Encryption Pack for access to these CSPs. These enhanced CSPs are available only to users living in countries to which the export of strong cryptography from the United States is permitted, and to organizations that are on the exception list of the U.S. cryptography export regulations. If you still have pre-SP2 Windows 2000 clients, you can download the High Encryption Pack for Windows 2000 from <http://windowsupdate.microsoft.com>. In Windows Server 2003 and XP, Microsoft also has added a new CSP that implements the Advanced Encryption Standard (AES) algorithm, the new U.S. standard for symmetric encryption.

Table 2-3 shows that Windows Server 2003 and XP contain both hardware and software implementations of CSPs. So far, Windows Server 2003 and XP include, by default, three smart-card vendor's CSPs: Infineon, Gemplus, and Schlumberger.

Microsoft received a Federal Information Processing Standard (FIPS) 140-1 Level 1 security certification for the following CSPs<sup>1</sup>: the Microsoft Enhanced Cryptographic Provider, the Microsoft Enhanced DSS and Diffie-Hellman Cryptographic Provider, and the Microsoft Enhanced RSA and AES Cryptographic Provider. If your IT environment requires a higher level of FIPS 140-1 compliance, have a look at the hardware-based, private-key storage solutions discussed at the end of this chapter.

---

<sup>1</sup> More information is available at <http://csrc.nist.gov/cryptval/140-1/1401val2002.htm>

**Table 2-3: Windows Server 2003 and XP Cryptographic Service Providers (CSPs)**

| CSP Name   | Description  |
|--|--|
| Microsoft Base Cryptographic Provider 1.0                        | Base CSP.  |
| Microsoft Base DSS Cryptographic Provider                        | Superset of the CSP in the previous row, including support for DSA and SHA.  |
| Microsoft Base DSS and Diffie-Hellman Cryptographic Provider     | Superset of the CSP in the previous row, including support for the Diffie-Hellman key agreement protocol.                                    |
| Microsoft Diffie-Hellman Schannel Cryptographic Provider         | Schannel CSP: used for secure Web communications using Secure Sockets Layer (SSL)/Transport Layer Security (TLS).                            |
| Microsoft RSA Schannel Cryptographic Provider                    |  |
| Microsoft Enhanced Cryptographic Provider 1.0                    | Enhanced version of base provider; supports longer key lengths. FIPS 140-1 Level 1 compliant.  |
| Microsoft Enhanced DSS and Diffie-Hellman Cryptographic Provider | Enhanced version of base provider; supports longer key lengths. FIPS 140-1 Level 1 compliant.  |
| Microsoft Enhanced RSA and AES Cryptographic Provider            | Enhanced version of base provider; supports longer key lengths and the AES algorithm for symmetric encryption. FIPS 140-1 Level 1 compliant. |
| Microsoft Exchange Cryptographic Provider 1.0                    | Exchange-specific CSP.   |
| Microsoft Strong Cryptographic Provider                          | Like the enhanced CSP, but it adds 40-bit encryption to make the strong CSP fully compatible with the base CSP.                              |
| Infineon SiCrypt Base Smart Card CSP                             | Infineon hardware CSP for smart-card support.  |
| Gemplus GemSAFE Card CSP 1.0                                     | Gemplus hardware CSP for smart-card support.   |
| Schlumberger Cryptographic Service Provider                      | Schlumberger hardware CSP for smart-card support.  |

## Certificate Templates

To let a CA cope with different certificate types, Microsoft has chosen to implement a flexible and modular architecture. The characteristics of a certificate—including the applications it can be used for—are defined in a certificate template stored in AD. Windows Server 2003 comes with support for version 2 certificate templates. Version 1 certificate templates were the ones Microsoft ships with Windows 2000 PKI. The key difference between version 1 and version 2 templates is that version 2 templates can be modified. Thanks to the support for version 2 templates, CA administrators can now also create their own templates, thus reflecting the certificate needs of the organization.

Certificate templates can also be used to define an enterprise CA's certificate issuance policy:

- You can use certificate templates to set the certificate types a CA can issue. You do this by loading the appropriate templates in the CA's policy module. To load the appropriate templates, use the Certification Authority Microsoft Management Console snap-in, right-click the Certificate Templates container, and select New\Certificate Template to Issue. You can also load templates from the command line using the certutil command with the -setCATemplates switch. The command below, for example, adds the user certificate template:

```
Certutil SetCATemplates +User
```

- On the Windows Server 2003 AD forest level, you can use certificate templates to set which users can both enroll and autoenroll for which certificate types. Like any other AD object, certificate templates have an ACL (list of permissions) that you can use to set which users can both enroll and autoenroll for a particular certificate type. To set the ACLs, use the Certificate Templates Microsoft Management Console snap-in (use the Security tab in the Properties of a certificate template).

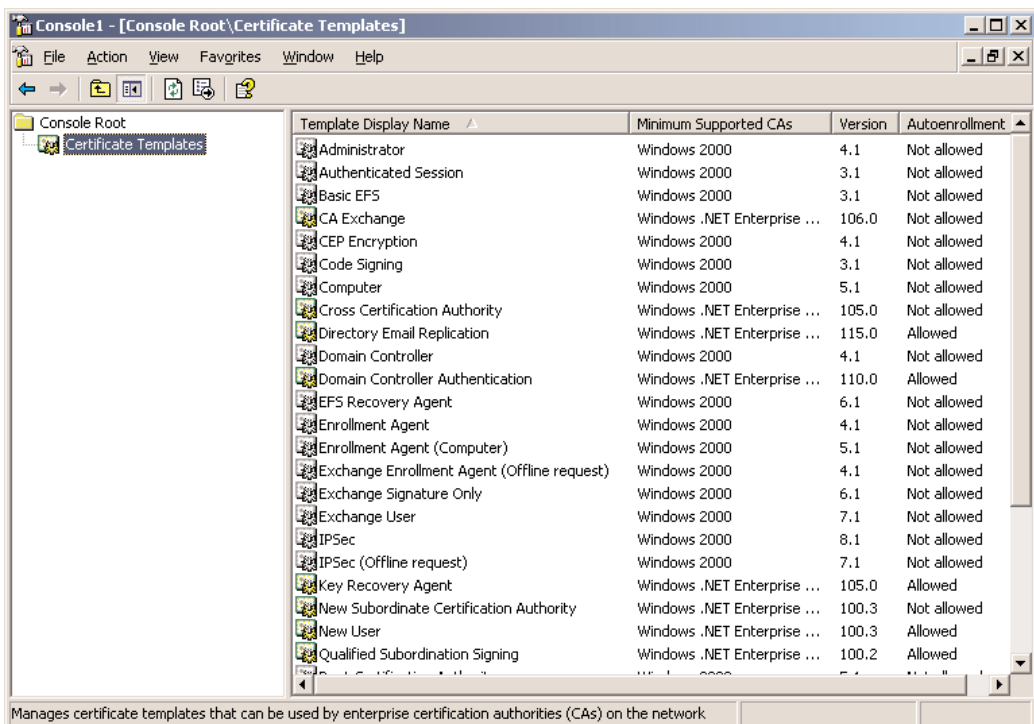
Because Windows standalone CAs lack AD integration, you cannot use certificate templates to customize a standalone CA's issuance policy.

Certificate templates are stored together with their properties in the system registry of servers that host a Windows Server 2003 CA (HKLM\Software\Microsoft\Cryptography\Certificatetemplatecache), and in the Active Directory Configuration naming context (cn=certificate templates, cn=public key services, cn=services, cn=configuration).

To administer version 2 certificate templates, use the Certificate Templates Microsoft Management Console snap-in (illustrated in Figure 2-5). A subset of a certificate template's definition is displayed in the Microsoft Management Console Certification Authority snap-in. Right-click a template and select Properties.

**Figure 2-5:**

*The Windows Server 2003 Certificate Templates Microsoft Management Console snap-in*



Support for version 2 certificate templates requires specific AD schema extensions. This is not a problem in a native Windows Server 2003 environment in which all domain controllers (DCs) are running the Windows Server 2003 Enterprise Server OS. If you have a mixed Windows Server 2003-Windows 2000 domain-controller environment, you can extend the AD schema using the `adprep.exe` utility (use the `/forestprep` switch), which is available from the Windows Server 2003 CD-ROM. Also, in this case at least, Windows 2000 Service Pack 3 will be required on the Windows 2000 DCs.

Certificates that are based on version 2 certificate templates cannot be issued from a CA running Windows Server 2003 Standard Edition. That's why, as mentioned earlier, it's always recommended that you install a CA using Windows Server 2003 Enterprise Edition or Data Center Edition.

From a PKI client point of view, all clients can enroll for certificates that are based on version 2 templates from a CA's Web-enrollment interface. Enrollment for a certificate that's based on a version 2 template from the Certificates Microsoft Management Console snap-in can be done only from a Windows XP or Windows Server 2003 OS platform. A user running the Windows 2000 OS can request certificates that are based on version 1 templates only from the Certificates Microsoft Management Console snap-in.

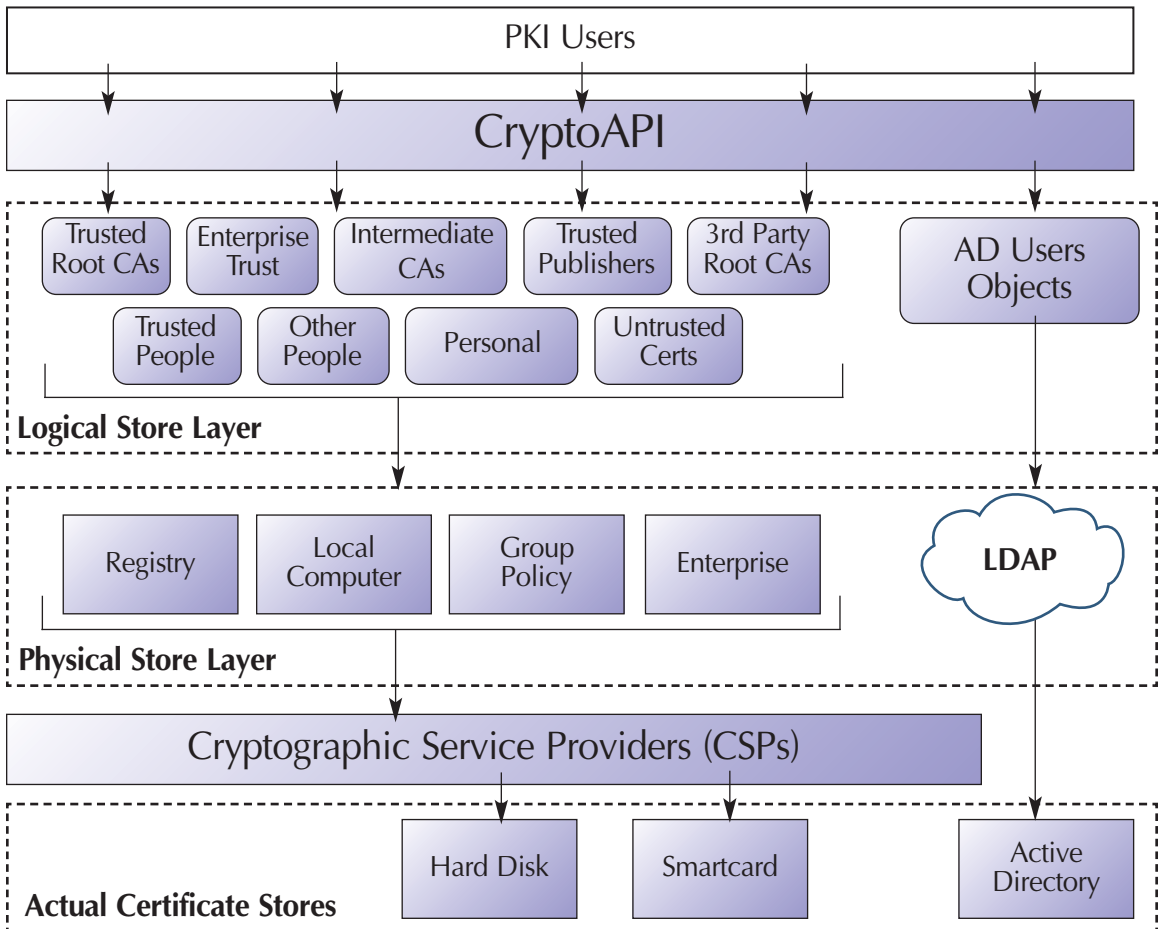
### Certificate Storage

In Windows, certificates are stored in a PKI entity's personal certificate store. In the paragraphs below, we go into more detail about the Windows Server 2003 and XP certificate stores architecture. Besides certificates, an entity's certificate store can also contain Certificate Trust Lists (CTLs), CRLs, and delta CRLs. Each Windows Server 2003 and XP user, machine, and service has its proper certificate store.

Figure 2-6 illustrates the certificate store architecture. As mentioned in the CryptoAPI section, one of the CryptoAPI's tasks is certificate management. CryptoAPI provides tools to attach certificates to messages and to store, retrieve, delete, list, and verify certificates.

The Windows Server 2003 and XP certificate store is divided into two abstraction layers: the logical certificate store and the physical certificate store. The purpose of this architecture is to abstract physical certificate storage from logical certificate categories. Windows PKI users shouldn't bother about where a certificate is physically stored, but rather about what they can do with it and to what category of PKI entity it belongs. The Windows certificate store architecture provides ways to make the content of multiple physical stores visible in one logical store, or, the other way around, to provide content inheritance from one physical store to multiple logical stores.

**Figure 2-6:**  
*Windows Server 2003 and XP physical and logical certificate stores*



To look at the content of a user, machine, or service certificate store and its different containers, use the Microsoft Management Console certificates snap-in. A user can also use the lightweight certificate viewer. The latter is accessible from the Internet Options/Content/Certificates menu in Internet Explorer.

By default, the Microsoft Management Console snap-in shows only the logical certificate containers. If you want, you can also see the physical certificate containers. To do this, select Options in the Microsoft Management Console View menu, and select the Physical certificate stores check box. To look at the user or machine certificate store from the command prompt, you can use the

certutil command-line utility. Use certutil with the -store switch to display the machine certificate store, and with the -user -store switches to display the user certificate store. Replace the -store switch with the -verifystore switch if you want to both list and verify the certificates in a store.

Windows Server 2003 and XP automatically archive expired and renewed certificates and their private keys in the certificate store. This process happens as part of the autoenrollment process. Archiving enables a user to decrypt old documents, even if the original certificate has expired or been renewed. To look at the archived certificates that are part of an entity's certificate store, select Archived certificates in the View options of the Microsoft Management Console certificates snap-in. The same menu contains a check box you can select to look at the certificates in an entity's certificate store, based on the certificates' purpose or application (as illustrated in Figure 2-7).

**Figure 2-7:**

*Classifying certificates in a certificate store based on certificate purpose*

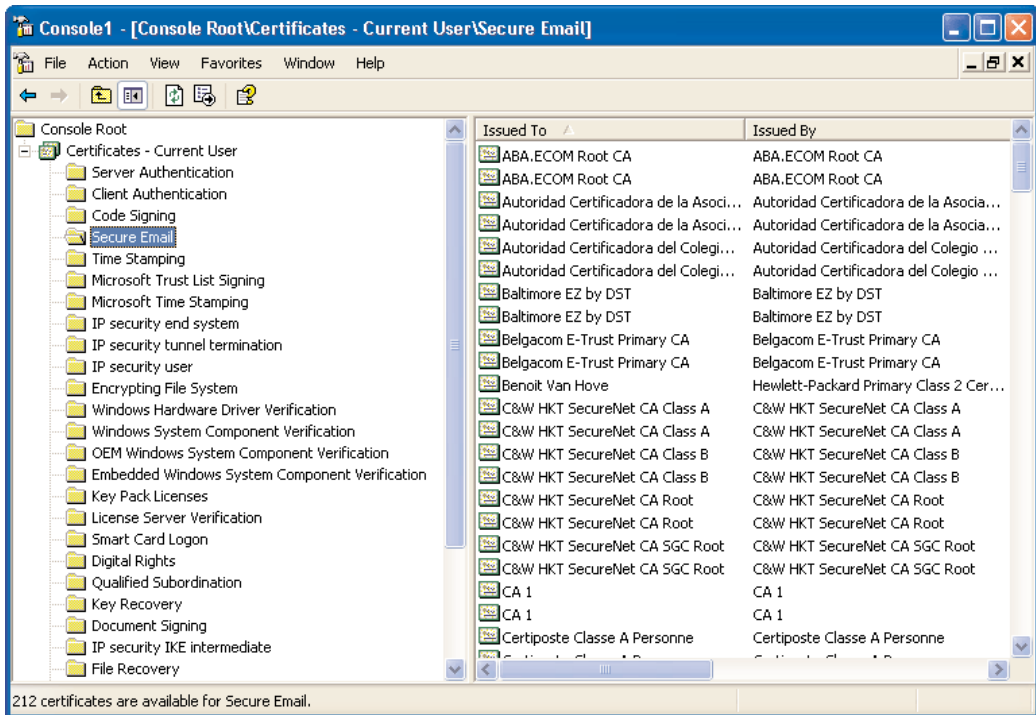


Table 2-4 contains an overview of the logical certificate containers and their meaning.

**Table 2-4: Overview of Logical Certificate Containers**

| Container Name                             | Meaning   |
|--|---|
| Personal                                   | Contains the certificates that are stored in a user's profile. Certificates in this folder have their corresponding private key stored in a secured user portion of the system registry.  |
| Trusted Root Certification Authorities     | Holds, by definition, only self-signed root CA certificates. In practice, however, also nonroot CA certificates can be added to this store. These CAs can be internal Windows 2000 CAs as well as CAs that are external to the company. This container is also referred to as the root store, which basically means that the certificates in this store are considered trust anchors. PKI trust and trust anchors are discussed in more detail in the next chapter of this eBook. |
| Enterprise Trust                           | Holds Certificate Trust Lists (CTLs). CTLs are signed lists that contain CA certificates.   |
| Intermediate Certification Authorities     | Holds, by definition, only intermediate CA certificates and their CRLs and delta CRLs. In practice, you also can add root CA certificates to this store. These intermediate CAs can be internal Windows CAs as well as CAs external to the company. The certificates in the Intermediate Certification Authorities store are NOT trust anchors.   |
| Active Directory User Object               | Holds user certificates that are published in the AD. This store exists only for user accounts—not machine or service accounts.   |
| Trusted Publishers                         | Holds Software Publisher certificates. These certificates are used to verify code that was signed using the Authenticode code-signing technology.   |
| Untrusted Certificates                     | Holds certificates that are not trustworthy. The presence of this container (that didn't exist in Windows 2000) is Microsoft's reaction to an incident that took place in 2001, in which a malicious person obtained a certificate for Microsoft's identity.  |
| Third-Party Root Certification Authorities | Theoretically holds only third-party root certificates and CRLs. In practice, you also can add root CA certificates to this store. As for the Trusted Root Certification Authorities container, the certificates in this container are, by default, considered trust anchors.   |
| Trusted People                             | Holds certificates of people who are explicitly marked as trustworthy.  |
| Other People                               | Holds the certificates of people with whom a user shares encrypted documents, such as S/MIME secured email messages.  |
| Certificate Enrollment Requests            | Holds certificate request files. These files are created when a user is requesting a certificate to a standalone Windows 2000 CA, or when an enterprise CA goes offline during a certificate enrollment request.  |

## Private Key Storage

In asymmetric cryptography—on which both PKI and PKI-enabled applications are built—access to and secure storage of the private key is critical. Because public keys are public entities, you shouldn't bother about secure public-key storage (this does not mean you shouldn't worry about the authenticity of the public key). Let's look at some examples of how you could misuse someone else's private key.

- If you gain access to someone's private email encryption key, you will be able to decrypt all the messages encrypted using the public key corresponding to that particular private key. If you get access to someone's email signing key, you will be capable of signing messages on that person's behalf.

- If you gain access to a CA's private key, you become even more powerful. With this key, you can issue fraudulent certificates and make CA users believe that you are the acting trusted third party. As such, you will have succeeded in compromising the complete CA trust system.
- If you gain access to a software vendor's code-signing private key, you can sign your code with the vendor's key. Users would then, without their knowledge, download and execute your (possibly malicious) code.

Private keys can be stored in different places. There are two main approaches to storing private keys: You can place them on dedicated hardware devices, or you can use software to hide them.

### **Software-Based Storage**

The bulk of today's operating systems—including Windows Server 2003 and XP—and PKI-enabled applications store private-key material in an encrypted format on the file system. Windows stores private keys or pointers to them (in case a hardware storage device is used) in the user's profile. The keys or pointers are stored in the subdirectory %UserProfile%\Application Data\Microsoft\Crypto\RSA. In Windows 2000, Windows Server 2003, and XP, the security of Microsoft's private-key storage on the file system is rooted in what Microsoft calls the Data Protection API (DPAPI). In previous OS versions, Microsoft used a technology that it referred to as the *protected store*.

### **Hardware-Based Storage**

Examples of dedicated hardware devices used for private-key storage are smart cards, USB tokens, and HSMs. A smart card or USB token is used to store the private key of a user. HSMs are used to store the private keys belonging to services or machines. A good example of HSM usage is secure CA private-key storage. Table 2-5 lists some popular vendors of these hardware devices (this list does not provide a complete overview).

**Table 2-5: Hardware Devices for Private-Key Storage: Solution and Vendor Overview**

| Vendors                                 | URL   |
|---|---|
| <b>Smart Card Vendors</b>               |   |
| Gemplus                                 | <a href="http://www.gemplus.com">http://www.gemplus.com</a>           |
| Schlumberger                            | <a href="http://www.schlumberger.com">http://www.schlumberger.com</a> |
| Oberthur                                | <a href="http://www.oberthurcs.com">http://www.oberthurcs.com</a>     |
| Datakey                                 | <a href="http://www.datakey.com">http://www.datakey.com</a>           |
| ActivCard                               | <a href="http://www.activcard.com">http://www.activcard.com</a>       |
| Spyrus                                  | <a href="http://www.spyrus.com">http://www.spyrus.com</a>             |
| <b>USB Token Vendors</b>                |   |
| eAlladin (eToken)                       | <a href="http://www.esafe.com">http://www.esafe.com</a>               |
| Rainbow Technologies (IKey)             | <a href="http://www.safenet-inc.com">http://www.safenet-inc.com</a>   |
| Spyrus                                  | <a href="http://www.spyrus.com">http://www.spyrus.com</a>             |
| <b>Hardware Security Module Vendors</b> |   |
| nCipher (nShield)                       | <a href="http://www.ncipher.com">http://www.ncipher.com</a>           |
| Safenet (Chrysalis-ITS Luna)            | <a href="http://www.safenet-inc.com">http://www.safenet-inc.com</a>   |
| Spyrus (Lynks)                          | <a href="http://www.spyrus.com">http://www.spyrus.com</a>             |
| Eracom (ProtectHost orange)             | <a href="http://www.era-com-tech.com">http://www.era-com-tech.com</a> |

Most dedicated hardware devices serve more goals than just to secure private-key storage. They can also provide secure on-board key-generation capabilities and cryptographic-processing capabilities to accelerate digital signing and other cryptographic operations. A major advantage of using external devices to store encryption and signing algorithms is that the private key never makes it to the user's PC.

Neither smart cards nor USB tokens are perfect, although both have excellent reputations in providing secure private key storage. Over the past few years, several researchers have conducted successful private-key chases on both types of devices. There are two important details in this context:

1. To conduct an attack on such a device, you need expensive and highly specialized hardware.
2. Most attacks require that you uncover the chip in the device. This requirement means that usually you cannot perform an attack without physically damaging the device.

A couple of successful USB token attacks are explained in detail at [http://www.atstake.com/research/reports/acrobat/usb\\_hardware\\_token.pdf](http://www.atstake.com/research/reports/acrobat/usb_hardware_token.pdf). For an overview of possible smart-card attacks, see the Cryptography Research website at <http://www.cryptography.com/dpa/technical/index.htm>.

To protect against physical tampering with a device, only one better category of solutions is available on the market: HSMs. But HSMs are expensive devices. That's why they are used only to protect very important keys, such as CA private keys. An interesting and relatively recent addition to HSM vendors' portfolios is network-attached HSMs. The key advantage of these devices is they can easily be shared between multiple applications that require advanced key protection or cryptographic acceleration. An example of a network-attached HSM is the Luna SA from Safenet.

When you evaluate hardware devices for secure private-key storage, make sure that you check their compliancy with important security standards such as the Common Criteria (<http://www.commoncriteriaportal.org/>), the U.S. government's FIPS 140-1 (<http://csrc.nist.gov/cryptval/>), and the U.K. government's ITsec (<http://www.cesg.gov.uk/site/iacs/index.cfm?menuSelected=1&displayPage=1>).