

Sponsored by

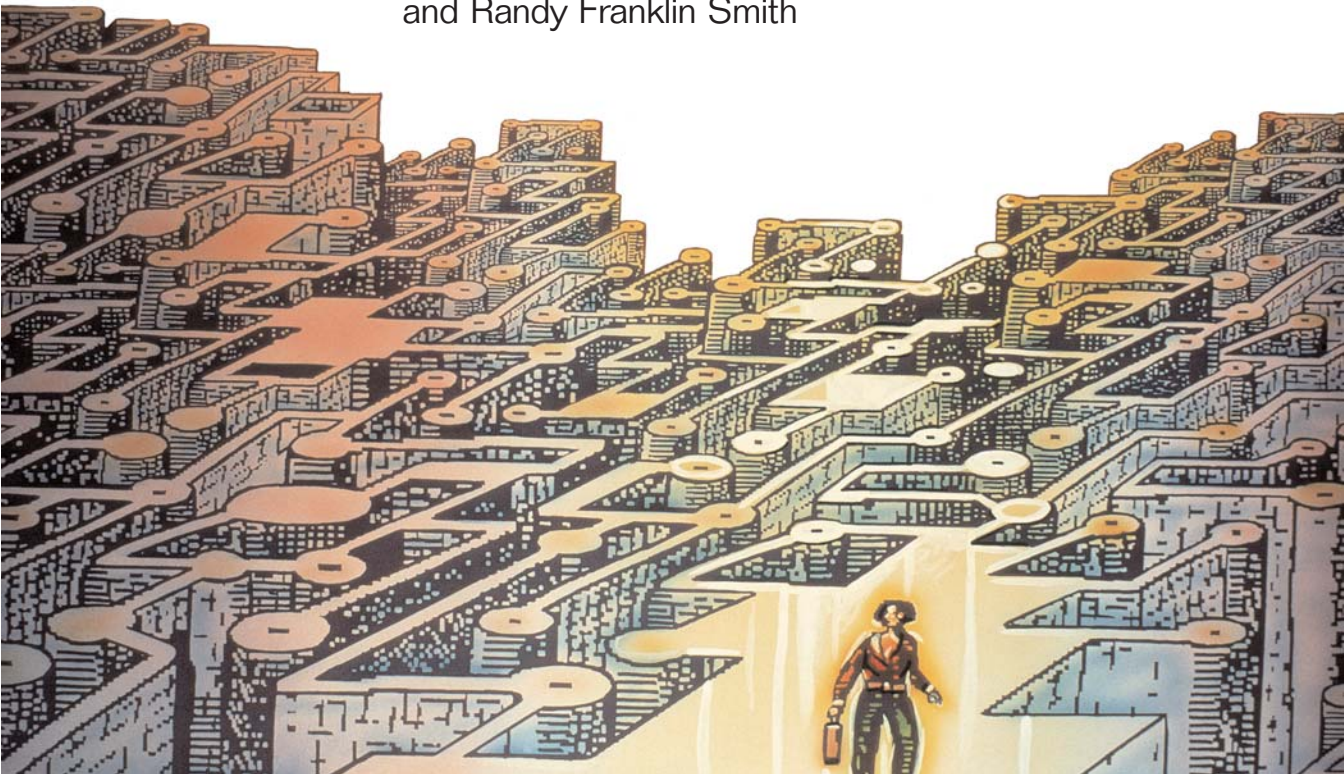


**ITPro**<sup>TM</sup> SERIES



A  
Guide to  
**Windows  
Certification  
& Public Keys**

By Jan De Clercq, Brett Hill, John Savill,  
and Randy Franklin Smith





## Contents

<b>Chapter 1: Uncover PKI and Certificate Services in Windows Server 2003</b> .....	<b>1</b>
<i>by Jan De Clercq</i>	
Windows 2003 Certificate Services Architecture .....	1
Windows 2003 Certificate Services Installation .....	5
Certificate Templates .....	7
Certificate Request Information Retrieval .....	7
Automated Certificate Enrollment .....	7
Centralized Key Archival .....	7
Certificate Request Approval .....	8
Publishing Certificates and CRLs .....	8
The Best CA for the Job .....	8
<b>Chapter 2: CA Trust Relationships in Windows Server 2003 PKI</b> .....	<b>9</b>
<i>by Jan De Clercq</i>	
Hierarchical Trust Model .....	9
The Networked Trust Model .....	11
Constrained Trust .....	12
Defining Trust Constraints .....	18
Flexible PKI Trust Definition .....	19
<b>Chapter 3: User-Side PKI Trust Management</b> .....	<b>20</b>
<i>by Jan De Clercq</i>	
User-Centric PKI Trust Management .....	21
Centralized User PKI Trust Management .....	23
Flexible PKI Trust Definition .....	25
<b>Chapter 4: Validating Digital Certificates in Windows PKI</b> .....	<b>26</b>
<i>by Jan De Clercq</i>	
Certificate-Validation Checks .....	26
Regular Certificate-Chain Processing .....	27
CTL Certificate-Chain Processing .....	30
Cross-Certification Chain Processing .....	30

**Chapter 5: Windows Server 2003 PKI Certificate Autoenrollment . . . . . 32**

*by Jan De Clercq*

How Autoenrollment Works . . . . .	32
Setting Up Certificate Autoenrollment . . . . .	34
Forcing Automatic Enrollment and Renewal . . . . .	36
Advanced Autoenrollment Options . . . . .	37
Ease of Use . . . . .	39

**Chapter 6: Understanding Windows PKI Certificate Revocation . . . . . 40**

*by Jan De Clercq*

Certificate Revocation Lists . . . . .	40
Revoking a Certificate . . . . .	43
PKI-Enabled Application Revocation Checking Support . . . . .	44
Automated Revocation Checking . . . . .	45
CRL Distribution Points . . . . .	45
Netscape Revocation Extensions . . . . .	48
A Crucial PKI Service . . . . .	48

**Chapter 7: Windows Server 2003 PKI Key Archival and Recovery . . . . . 49**

*by Jan De Clercq*

Configuring Automatic Key Archival and Recovery . . . . .	49
Automatic Key Archival and Recovery Architecture . . . . .	51
<i>Sidebar: Manual Key Archival and Recovery</i> . . . . .	52
Key Recovery . . . . .	53
Data Recovery vs. Key Recovery . . . . .	55
Powerful Capabilities . . . . .	56

**Chapter 8: Using Certificates to Secure Your WLAN . . . . . 57**

*by Randy Franklin Smith*

Adding X . . . . .	58
Certificate Services . . . . .	58
Obtaining Certificates . . . . .	59
Wireless Client Network Settings . . . . .	59
Configure IAS and the APs . . . . .	61
Test Case . . . . .	62

<b>Chapter 9: FAQs</b> .....	<b>63</b>
Obtaining a Server Certificate from Your Own CA .....	63
<i>by Randy Franklin Smith</i>	
Using Windows Server 2003's Certificate Templates .....	64
<i>by Randy Franklin Smith</i>	
Enabling SSL on Your Site .....	65
<i>by Brett Hill</i>	
Using the SSL Protocol to Secure HTTP Basic Authentication Traffic .....	66
<i>by Jan De Clercq</i>	
Addressing ActiveX Controls .....	66
<i>by John Savill</i>	
Enabling SSL on IIS .....	67
<i>by John Savill</i>	
IIS Client Service Mapping .....	67
<i>by Jan De Clercq</i>	
Controlling Which CAs Windows Can Trust .....	70
<i>by Randy Franklin Smith</i>	
Mitigating a Problem with Computer-Only Authentication to a WLAN .....	71
<i>by Randy Franklin Smith</i>	
Setting Up SSL Certificates for an NLB Cluster .....	72
<i>by Paul Robichaux</i>	

## Chapter 1

# Uncover PKI and Certificate Services in Windows Server 2003

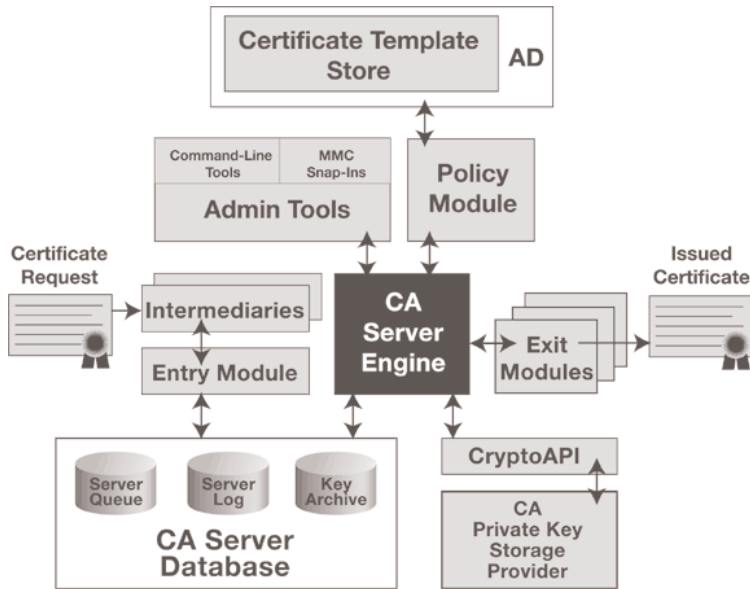
—by *Jan De Clercq*

The core component of the Windows Server 2003 public key infrastructure (PKI) software is the Certification Authority (CA), which Microsoft often refers to as the Certificate Server or Certificate Services. A CA receives and processes PKI user certificate requests, identifies and validates those requests, issues certificates according to the PKI's security policy, renews and revokes certificates, publishes certificates to different locations, creates and publishes certificate revocation lists (CRLs), and logs all certificate and CRL transactions to the appropriate database. A Windows 2003 CA can also perform secure private key archival and recovery. To better understand how CAs and PKI have evolved in Windows 2003, let's examine the components of the latest Certificate Services architecture and the differences between establishing an enterprise CA and a standalone CA in Windows 2003.

## Windows 2003 Certificate Services Architecture

The Windows 2003 Certificate Services architecture is almost identical to the architecture that Microsoft used for previous editions of Certificate Services. A key difference is that Microsoft modified the CA database layout to let the CA archive and recover PKI users' private keys. Figure 1 shows the architecture, which includes various modules, databases, administrative tools, intermediaries, and CryptoAPI.

**Figure 1**  
*Certificate Server architecture*



**Modules.** At the heart of Certificate Services sits a CA server engine (`certsrv.exe`) that generates certificates and CRLs and directs the message flow between the CA and other Certificate Services components. The engine uses the entry, policy, and exit modules to communicate with the other components.

The entry module accepts certificate requests formatted according to Public-Key Cryptography Standards (PKCS) #10 or the Cryptographic Management protocol using Cryptographic Message Syntax (CMS). After accepting the requests, the entry module places them in a queue for processing by the policy module.

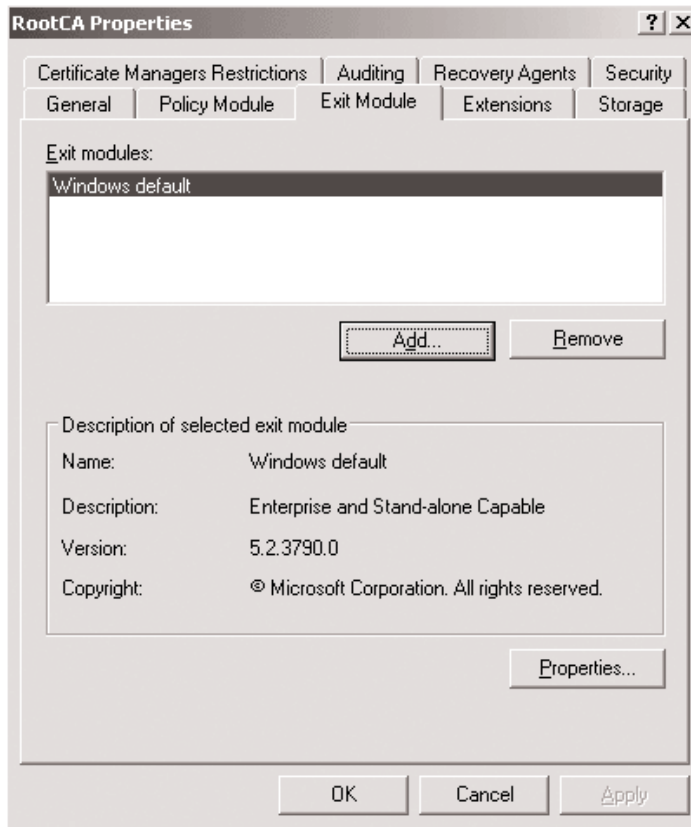
The policy module implements and enforces the CA policy rules as set by the CA administrator. The policy module informs the CA server engine about the layout of a certificate and decides whether the CA should issue a certificate, deny a certificate, or leave a certificate request pending. To retrieve certificate layout information, the policy module can call on information stored in a directory (e.g., Active Directory—AD) or database. Windows 2003 comes with a policy module called `certpdef.dll` that supports two policy types: the enterprise policy mode and the standalone policy mode. (I discuss these policy modes in greater detail later.) To check out the policy module installed on your CA, open the Microsoft Management Console (MMC) Certification Authority snap-in, right-click the CA object, select Properties from the context menu, and select the Policy Module tab, which Figure 2 shows.

**Figure 2**  
*Policy Module Tab*



Exit modules distribute and publish certificates, certificate chains, complete CRLs, and delta CRLs. Exit modules can write PKI data to a file or use HTTP or a remote procedure call (RPC) to transport the data to a remote location. Because a Windows 2003 CA can support multiple exit modules, the CA can publish and distribute certificates, certificate chains, complete CRLs, and delta CRLs to different locations at the same time, including Lightweight Directory Access Protocol (LDAP) directories, file shares, Web directories, and even ODBC-compliant databases. The default Windows 2003 CA exit module is called certxds.dll and comes with LDAP, FTP, HTTP, and SMTP support (the Windows 2000 CA supports all but the last protocol). Exit modules let the CA automatically send notification messages to PKI users and administrators. To check out the exit modules installed on your CA, open the Certification Authority snap-in, right-click the CA object, select Properties from the context menu, and select the Exit Module tab, which Figure 3 shows.

**Figure 3**  
*Exit Module tab*



The policy module and the exit modules are both customizable and replaceable—any organization can develop its own modules in C++ or Visual Basic (VB) and plug them into the Certificate Services architecture. The Windows 2003 platform software development kit (SDK) documents the steps necessary for creating and replacing modules.

You can use the Certification Authority snap-in or the certutil.exe command-line utility to configure the policy and exit modules. By using the properties associated with the CA object in the Certification Authority snap-in, you can add multiple exit modules, configure X.509 certificate extensions (e.g., CRL Distribution Points—CDP—and Authority Information Access—AIA—points), and configure complete CRL and delta CRL publication parameters.

**Databases.** The CA has its own database, called *CAname.edb*, to store certificate transactions and status information, certificates, and optionally archived private keys. By default, the database is in the `\%systemroot%\system32\certlog` folder. The CA engine communicates with its database through the `certdb.dll` file. With the release of Win2K Certificate Services, Microsoft changed its database technology to the Jet database engine and in doing so made the Win2K CA scalable. Microsoft uses the same technology for the AD and Microsoft Exchange Server databases.

**Administrative tools.** Although most administrators will use the Certification Authority snap-in to manage a Windows 2003 CA, you can also use the `certutil.exe` command-line utility. Both administration tools use the `certadm.dll` file to communicate with the CA engine.

**Intermediaries.** Applications that help the client generate correctly formatted PKCS #10 or CMS certificate request files are known as intermediaries or registration authorities (RAs). An intermediary or RA gathers user- and request-specific data required for a valid certificate request. For example, any request sent to a Windows 2003 enterprise CA should mention a certificate template. An intermediary can add a template specification to the request. Intermediaries are bound to a specific transport protocol (e.g., HTTP, RPC). As a result, the CA engine doesn't have to work with different transport providers.

Examples of Windows 2003 intermediaries are the Web-based certificate enrollment pages, which serve as an HTTP intermediary, and the MMC Certificates snap-in that calls on the Certificate Request Wizard, which is an RPC intermediary. The HTTP intermediary calls on 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 on the `certcli.dll` file to perform these tasks.

**CryptoAPI.** For all cryptographic functions, including accessing and using the CA's private key, the CA calls on CryptoAPI. The CA can store its private key on a hard disk or on a dedicated hardware device (e.g., a Hardware Security Module—HSM).

## Windows 2003 Certificate Services Installation

When you install Windows 2003 Certificate Services, you can install a root CA, a subordinate CA, an enterprise (AD-integrated) CA, or a standalone (nonAD-integrated) CA. Installing Certificate Services in enterprise mode activates the enterprise mode of the Windows 2003 CA policy module. Let's compare the enterprise mode with the standalone mode to see how they differ. Table 1 compares the default characteristics of a Windows 2003 standalone CA with a Windows 2003 enterprise CA.

**Table 1 Windows 2003 Standalone CA vs. Windows 2003 Enterprise CA**

Feature	Windows 2003 Standalone CA	Windows 2003 Enterprise CA
AD integration	By default, nonAD-integrated.	AD-integrated.
CA communication protocols	Communication with the CA front end occurs across HTTP or HTTP Secure (HTTPS).	Communication with the CA front end can use RPC or Distributed COM (DCOM) or HTTP or HTTPS.
Certificate distribution	The CA downloads the certificate to the user profile when the user manually retrieves the certificate from the CA Web site. You can automatically publish CRLs and certificates to AD.	Depending on the certificate template, the CA automatically downloads the certificate to the user profile, publishes it in AD, or both.
Certificate request approval	Certificate enrollment approval can be automatic or manual. The CA has one setting that controls this behavior for all certificate types.	Certificate enrollment approval can be automatic or manual. You can control this behavior globally at the CA level or per certificate by using a certificate template setting. Also, the certificate approval process can use the AD authentication and access control model through the access control lists (ACLs) that are set on certificate templates.
PKI user scope	Extranet and Internet PKI user-oriented.	Intranet PKI user-oriented.
Platform requirements	You can install this version on a Windows 2003 DC, member server, or a standalone server (i.e., not a member of any domain).	You can install this version on a Windows 2003 DC or member server.
Supported certificate types	Can issue a limited set of certificate types and certificates that require a custom OID in their Extended Key Usage extension; doesn't support certificate templates.	Can issue all Windows 2003 certificates defined in the Windows 2003 Certificate Templates snap-in; supports version 1 (Win2K PKI) and version 2 (Windows 2003 PKI) certificate templates.
User identification	Users must enter identification information manually when requesting a certificate.	The CA automatically retrieves user identification information from AD.
User management	Users enroll by using a Web-based interface. You can also use the certreq.exe command-line utility.	Users enroll by using a Web-based interface or the Certificates snap-in. You can also use the certificate autoenrollment feature to enroll users automatically or the certreq.exe command-line utility.

To install a CA in enterprise mode, the account installing the CA must be an enterprise administrator and a domain administrator of the AD forest's root domain. In addition, the server on which you install the enterprise CA must be a member of a domain that has a functioning AD. If these conditions aren't met, the enterprise mode installation options will be shaded in the CA Installation Wizard and you'll be able to install the CA only in standalone mode.

To install a CA in standalone mode, no AD is required. You can install the CA on a standalone server, a member server, or a domain controller (DC). Also, the account performing the installation doesn't need to be an enterprise or domain administrator—local machine administrator permissions are sufficient. If you do use enterprise administrator privileges to install a standalone CA, the CA will

offer some additional features. For example, if an enterprise administrator installs a standalone CA on a member server that's joined to the domain, the CA will publish to the AD the certificates that it issues.

## Certificate Templates

An enterprise CA uses certificate templates stored in the AD configuration naming context (NC). Certificate templates define the content and characteristics of a certificate. Certificate templates also provide a way to control which certificate types an enterprise CA can issue and which users can request which certificate types from an enterprise CA. Windows 2003 PKI supports version 2 certificate templates. In contrast to version 1 templates, version 2 templates are fully customizable. You can use the MMC Certificate Templates snap-in to customize them.

A standalone CA can't use AD certificate templates. As a consequence, you can't control which users can request which certificate types from the CA. By default, a standalone CA can issue only Web authentication (Secure Sockets Layer—SSL—or Transport Layer Security—TLS), email protection (Secure MIME—S/MIME), server authentication, code signing, timestamp signing, and IP Security (IPSec) certificates. You can, however, modify the standalone CA's Web interface (e.g., to list other certificate types) or simply request other certificate types by using special Object Identifier (OID) values stored in a certificate's Extended Key Usage X.509 extension.

## Certificate Request Information Retrieval

An enterprise CA retrieves user information from AD during certificate enrollment. The CA uses this information to populate certain certificate fields. For example, a certificate issued by an enterprise CA contains a reference to a user's user principal name (UPN) in the certificate's SubjectAltName X.509 field. Because a standalone CA has no access to AD, the user must manually complete the user identification information required for the certificate on the enrollment Web site.

For both enterprise and standalone CAs, you can change the default values that the CA adds to a certificate request at the time of enrollment by editing the `certdat.inc` file in the `%systemroot%\system32\certsrv` directory. You can change the default values for the following certificate entries: `sDefaultCompany`, `sDefaultOrgUnit`, `sDefaultLocality`, `sDefaultState`, and `sDefaultCountry`. Changing these values to your organization's default values reduces the amount of information that a user must enter when requesting a certificate.

## Automated Certificate Enrollment

An enterprise CA supports automated certificate enrollment, and Windows 2003 extends this feature to cover both users and machines. For additional information about automated certificate enrollment, see Chapter 5, "Windows Server 2003 PKI Certificate Autoenrollment." A user who wants a certificate from a standalone CA must manually start the enrollment process—no automation is provided.

## Centralized Key Archival

Whereas an enterprise CA supports centralized key archival in the CA database, a standalone CA doesn't. For more information about the CA's key archival and recovery capabilities, see Chapter 7, "Windows Server 2003 PKI Key Archival and Recovery."

## Certificate Request Approval

An enterprise CA can support automatic or manual certificate request approval. You can use the properties of an enterprise CA (using the policy module properties) or the properties of a version 2 certificate template (using the Issuance Requirements tab in the certificate template's properties) to set the certificate request-handling properties. If you use a version 2 certificate template to set the certificate request-handling properties, the change will apply only to the certificate type defined in the template. You can force the administrator to manually approve all incoming certificate requests— independent of the certificate template settings—by setting the request-handling properties to *Set the certificate request status to pending. The administrator must explicitly issue the certificate* in the CA properties. The same options are available for a standalone CA. The only difference is that a standalone CA can't use certificate templates, and as a consequence, you can't set request-handling properties for individual certificate templates. Unlike an enterprise CA, a standalone CA doesn't rely on the built-in Windows authentication mechanisms to authenticate incoming certificate requests; therefore, Microsoft doesn't recommend setting a standalone CA's request-handling property to automatic approval. You should always leave this property set to the default to require manual CA administrator approval for every incoming certificate request.

## Publishing Certificates and CRLs

An enterprise CA uses AD to store and publish certificates, complete CRLs, and delta CRLs. Both a standalone CA and an enterprise CA can also publish to the file system. Each certificate published in AD automatically maps to the Windows account of its requestor. AD adds the certificate to the multivalued userCertificate attribute of a user or inetOrgPerson AD object. However, not every certificate that an enterprise CA generates is automatically published in AD. Examples of certificates that aren't automatically published are an enrollment agent or certificate trust list (CTL) signing certificate.

A standalone CA can publish issued certificates to AD, but this step isn't the default behavior. A standalone CA will automatically publish certificates to AD only if an enterprise administrator installs the CA on a member server joined to the domain. You can obviously always publish the certificates manually to AD.

## The Best CA for the Job

Now that you're aware of the differences between an enterprise CA and a standalone CA, you can pick the best option for your situation. A Windows 2003 enterprise CA typically is best suited for enterprise certificate users who have an AD user account and who use Kerberos to authenticate to the AD infrastructure. A Windows 2003 standalone CA typically is best suited for external users (e.g., extranet users) who don't have an internal Windows account.

## Chapter 2:

# CA Trust Relationships in Windows Server 2003 PKI

—by Jan De Clercq

The most fundamental question in a public key infrastructure (PKI) is, “Which public keys are trustworthy?” The answer to that question starts with trust in a Certification Authority (CA). When you trust a CA, you expect that CA to create legitimate certificates that uniquely bind information about an individual to a public key. You also expect the CA to verify the individual’s identity and determine whether his or her private key is stored securely—before the CA issues a certificate.

PKI trust models provide a technological framework for managing the trust relationships between CAs and PKI users and between CAs. A CA’s trust domain defines the organizational or geographical boundaries within which the CA is considered trusted. One organization might be divided into different trust domains according to, for example, the organization’s divisions or departments. All PKI users in a CA’s trust domain consider the CA a trust anchor—a CA that the PKI user explicitly trusts under all circumstances. During certificate validation, PKI software tries to discover a trust path that reaches to the level of a trust anchor CA.

Windows Server 2003 PKI supports two main PKI trust models: the hierarchical trust model and the networked trust model. Windows 2003 PKI also features built-in support for constrained PKI trust relationships, which let CA administrators qualify trust relationships between CAs.

## Hierarchical Trust Model

The hierarchical PKI trust model, which consists of a tree of CAs, is typically used to define PKI trust relationships within one organization. Organizations that have a clear hierarchical structure can often easily be mapped to this type of model.

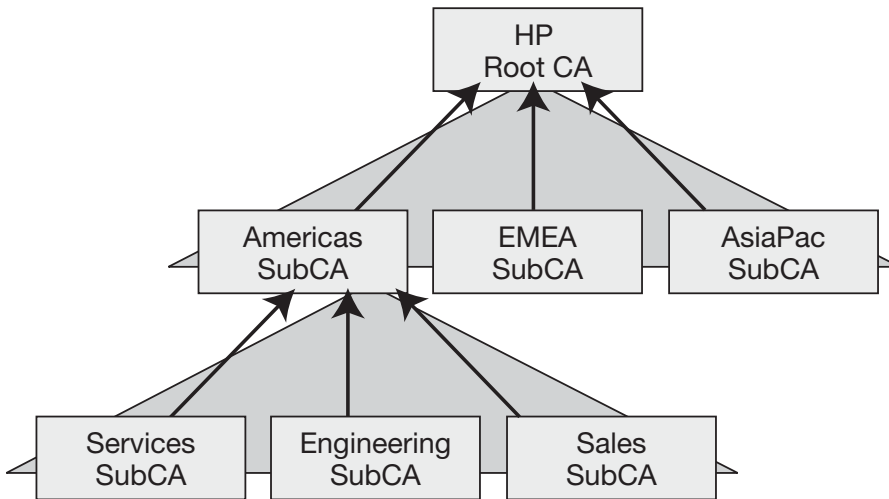
In a hierarchical trust model, a clear superior-subordinate relationship exists between CAs on different hierarchical tiers. At the top of the hierarchy is a root CA—typically the trust anchor of the hierarchy. Within a hierarchy, the root CA is the only entity authorized to sign its own certificate. The self-signed root certificate makes it impossible for anyone to pretend to be the root CA; only the root CA knows and possesses its private key. The root CA certifies the tier-1 CAs (one tier below the root), which in turn certify the tier-2 CAs, and so on.

Figure 1 shows a sample hierarchical PKI trust model comprising three levels of CAs: the root CA level, an intermediate CA level (based on the organization’s geographical locations), and an issuing CA level (based on the organization’s divisions). The hierarchical trust model supports delegation: A superior CA can delegate part of its certificate-issuing responsibilities to a subordinate CA. In a *strict* hierarchy, subordinate CAs (i.e., non root CAs) have only one superior CA. A hierarchy can contain

two types of subordinate CAs: issuing and intermediate. Issuing CAs issue certificates to PKI users, whereas your intermediate CAs should (as a best practice) issue certificates only to subordinate CAs. This approach lets you take intermediate CAs offline to provide an additional level of CA security.

**Figure 1**

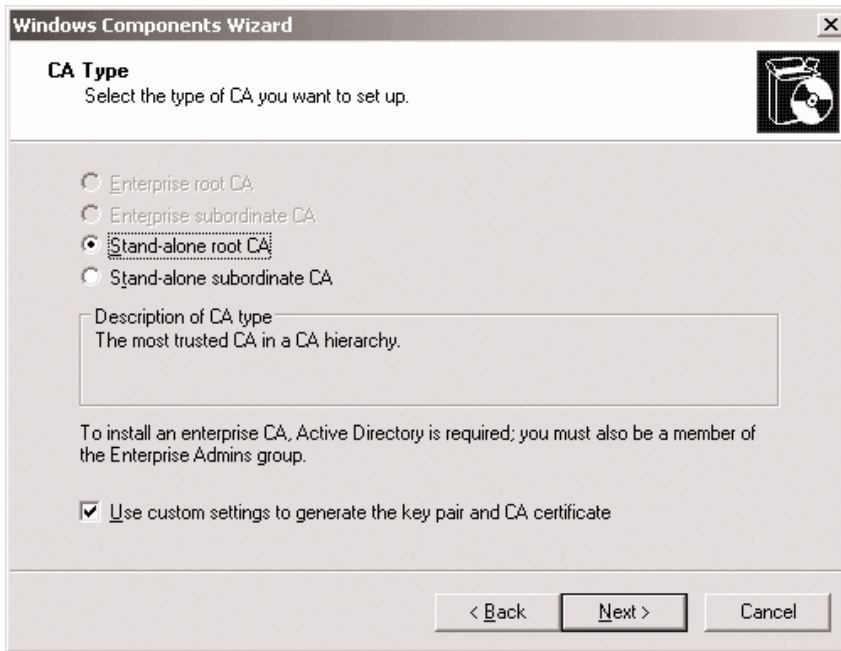
*Hierarchical trust model*



When validating a certificate issued by a CA that's part of a hierarchical trust model, the PKI client software attempts to discover a trust path that links the issuing CA to the CA trust anchor. Windows 2003, Windows XP, and Windows 2000 PKI clients all support the necessary trust-path discovery and traversal mechanisms in a hierarchical trust model.

You define a hierarchical trust relationship between a superior CA and a subordinate CA during the Windows 2003 CA installation process. The Windows Components Wizard's CA Type dialog box, which Figure 2 shows, lets you select whether the CA will be a root CA or a subordinate CA. When building a CA hierarchy, you should use standalone CAs for the root CA and intermediate (i.e., nonissuing) CAs. Doing so will facilitate taking these CAs offline. The issuing CAs can be Windows 2003 enterprise CAs, which are integrated with Active Directory (AD).

**Figure 2**  
*Windows Components Wizard CA Type dialog box*



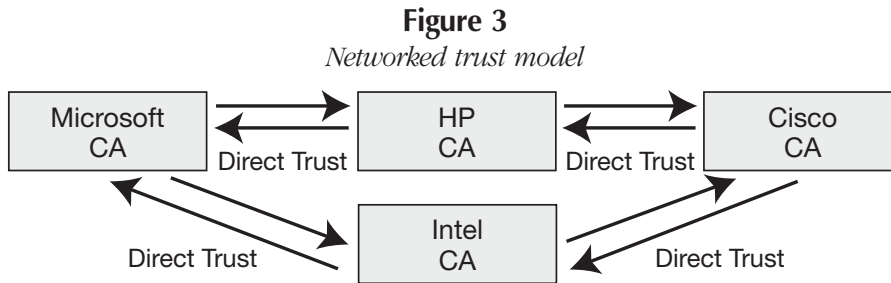
For a subordinate CA, the CA installation process offers two choices. The first choice is to submit a certificate request, which is a \*.req file that contains a Public-Key Cryptography Standards #10 (PKCS #10) or Certificate Management protocol with Cryptographic Message Syntax (CMS)-formatted request blob, directly to the parent CA (if the CA is published in AD and online). The second choice is to provide the request to the parent CA manually (e.g., using a 3.5" disk).

## The Networked Trust Model

A networked trust model (aka a peer-to-peer—P2P—model or distributed trust model) has no superior-subordinate relationships between different CAs—all CAs are considered peers. The networked trust model is typically used to define PKI trust relationships between organizations. You can choose one of two methods—certificate trust lists (CTLs) or cross-certification—to set up trust relationships in a P2P model. Windows 2003 PKI supports both methods, whereas Win2K PKI supports only CTLs.

A CTL is a signed list of trusted CA certificates that's centrally managed by a PKI administrator and distributed throughout the organization to all PKI clients. You use a Group Policy Object (GPO) setting to define CTLs. In Windows 2003 PKI and Win2K PKI, you can limit a CTL's validity period and you can configure its scope to a limited number of PKI-enabled applications.

Cross-certification simply means that a CA issues a certificate to and can receive a certificate from another peer CA. A cross-certification trust relationship can be one-way or two-way (i.e., the CAs cross-certify each other). Figure 3 shows a networked PKI trust model set up between organizations.



Windows 2003 PKI clients and XP PKI clients support the necessary trust-path discovery and traversal mechanisms for a networked trust model made up of several cross-certifications. These mechanisms aren't available on Win2K PKI clients. When validating a certificate issued by a CA that's part of a networked trust model, the PKI client software will try to discover a trust path that links the issuing CA to its local CA trust anchor.

Contrary to setting up a hierarchical trust relationship, you can set up a cross-certification trust relationship any time after the CA installation. However, you can't set up a cross-certification trust relationship from the Windows PKI graphical interface; you must do so from the command line by using the `certreq.exe` utility. The instructions for setting up a cross-certification trust relationship are available in the Windows 2003 product documentation and in the Microsoft white paper at <http://www.microsoft.com/technet/prodtechnol/windowsserver2003/technologies/security/ws03qswp.msp>.

## Constrained Trust

Windows 2003 PKI's support for constrained PKI trust relationships, or what Microsoft calls *qualified subordination*, lets CA administrators put constraints on trust relationships between a CA and its subordinate CAs (in a hierarchical trust model) or between peer CAs (in a networked trust model). This ability to qualify trust relationships aligns PKI trust more closely with real-life trust: In reality, trust is rarely complete and is usually subject to certain conditions.

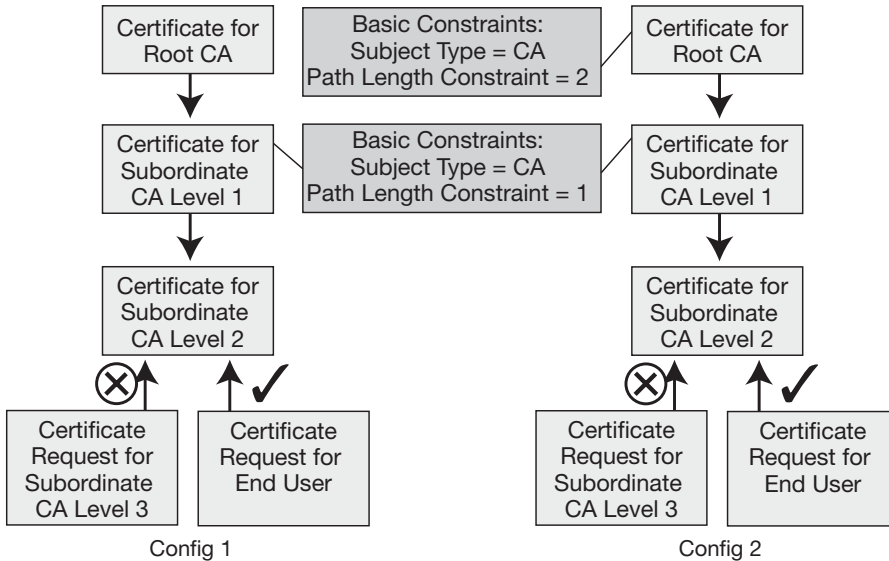
You define trust constraints by embedding specific X.509 certificate extensions in a subordinate CA's certificate or a peer cross-certified CA's certificate. Windows 2003 PKI supports the basic, name, issuance policy, and application policy trust-constraint-related certificate extensions.

**Basic constraints.** Basic constraints are based on an X.509 certificate extension called Basic Constraints, which can contain a field called `pathLenConstraint` (or path-length constraint). You can use this field only when the Basic Constraint X.509 certificate extension's `ca` field is set to `true`—which is the case only for a CA certificate. The path-length constraint sets the maximum number of non-self-issued CA certificates that can follow a certificate in a certification path, so you can use it to limit the length of the certificate chain.

Figure 4 shows two sample hierarchical trusts: Config 1 and Config 2. In Config 1, a basic constraint in the Level 1 subordinate CA's certificate limits that certificate's path length to 1. As a consequence, the CA located one level below that CA can't issue CA certificates, only end-user certificates. Config 2 obtains the same result by adding a path length constraint of 2 to the root CA's certificate. In that case, the PKI software automatically adds a path-length constraint of 1 to all subordinate CA certificates that the root CA issues.

**Figure 4**

*Basic constraint example*



**Name constraints.** You can set a name constraint certificate extension only in a CA certificate. The name constraint lets you restrict the population to which a subordinate or cross-certified CA can issue certificates. You can use name constraints to specify a namespace within which the subject names and subject alternate names in a certificate request, or the IP addresses that issued the certificate request, must be located. The constraints reside in the CA certificate's Name Constraints extension. Table 1 lists the name constraint types that Windows 2003 PKI supports. The table also lists the Internet Engineering Task Force (IETF) Request for Comments (RFC) standards that define each type.

**Table 1 Name Constraint Types**

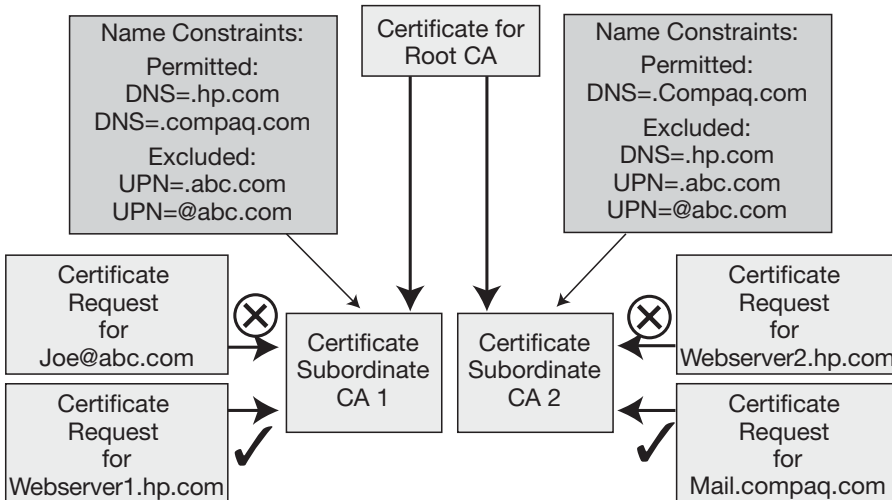
Name Constraint Type	Based On	Function
DNS names	RFC 1035 and RFC 1034	Restricts certificate issuance based on the DNS name in the certificate request
Email and UPN	RFC 822	Restricts certificate issuance based on the email address or UPN in the certificate request
IP address	RFC 2460 or RFC 791	Restricts certificate issuance based on the IP address of the machine from which the certificate request was sent
Uniform Resource	RFC 2396	Restricts certificate issuance based on the URI Identifiers (URIs) in the certificate request; can be used to limit the issuance of SSL Web server certificates
X.500 distinguished	X.500	Restricts certificate issuance based on the DN names (DNs) in the certificate request; can be used to issue certificates to a limited number of users or computer objects in AD

In accordance with RFC 3280, a certificate Name Constraint extension specifies a namespace within which all subject names in subsequent certificates in a certification path must be located. As a consequence, a subordinate CA can only reduce—never extend—the namespace rule it receives from its parent CA. For example, if a subordinate CA is permitted to issue certificates for users in the research.hp.com DNS domain, that CA can never issue a subordinate CA certificate that permits that subordinate CA to issue certificates for users in the hp.com DNS domain.

In the hierarchical trust example that Figure 5 shows, you see that a name constraint can contain both exclusive and inclusive rules. During name constraint validation, exclusive rules always have precedence over inclusive rules. In the example, a name constraint in subordinate CA 1's certificate excludes the user principal name (UPN) @abc.com and permits the DNS names .hp.com and .compaq.com. When CA 1 receives a certificate request for joe@abc.com, it will reject the request. If a request comes in for webserver1.hp.com, CA 1 will accept the request. Subordinate CA 2's namespace is even more restricted: CA 2 can issue certificates only in the .compaq.com DNS namespace. As a consequence, when CA 2 receives a request for webserver2.hp.com, it rejects the request. If CA 2 receives a request for mail.compaq.com, it will accept the request.

**Figure 5**

*Name constraint example*



**Issuance policy constraints.** An issuance policy defines the conditions that were met when the certificate was issued. In a Windows 2003 certificate, an issuance policy is identified by using its corresponding Object Identifier (OID) and is kept in a certificate's *Certificate policies* extension.

When you add issuance policy constraints to a CA certificate, they define the set of issuance policies that will be included in any certificate that the CA issues. You can include the constraints in cross-certification CA certificates to limit the trust you have in the certificates that a cross-certified CA issues. The enforcement of an issuance policy constraint extension relies on the certificate chain validation logic. This logic is available only in Windows 2003 and XP.

When you include issuance policies in a certificate issued to a PKI user, policy enforcement must be performed at the level of the PKI-enabled application. The PKI-enabled application must check to determine whether it permits a certificate issued under a certain issuance policy. Thus, the application must be intelligent enough to know which issuance policies it supports.

Windows 2003 PKI comes with four predefined issuance policies. Table 2 shows each policy's corresponding OID and function. The a, b, c, and d variables in the OID for the low, medium, and high assurance issuance policies represent a randomly generated value that's unique for every Windows 2003 forest. You can also define your own policies, depending on the needs of your PKI environment or application.

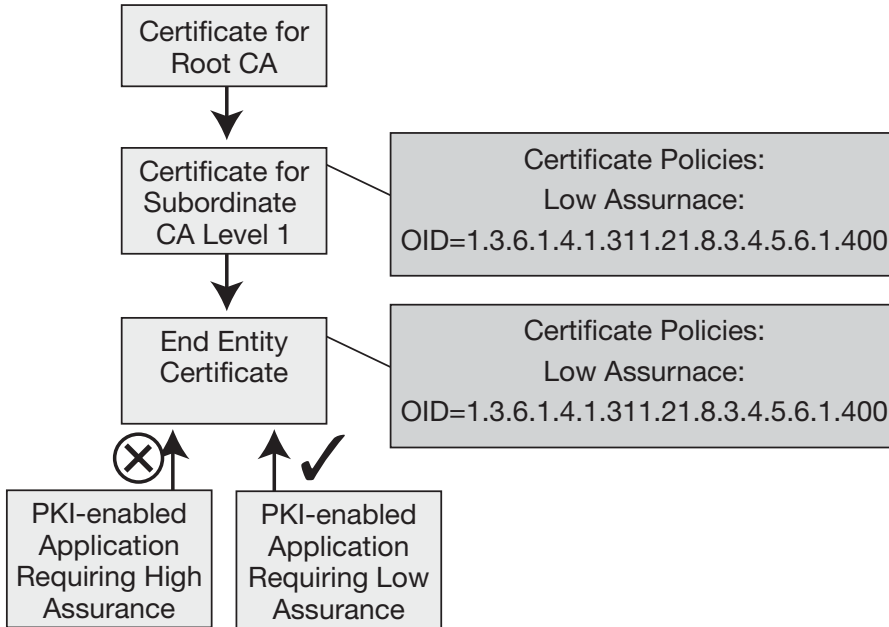
**Table 2 Predefined Windows 2003 PKI Issuance Policies**

Issuance Policy Type	OID	Function
All issuance	2.5.29.32.0	Issuance policy containing all other issuance policies; used only for CA certificates
Low assurance	1.3.6.1.4.1.311.21.8.a.b.c.d.1.400	Certificates issued with no additional security requirements
Medium assurance	1.3.6.1.4.1.311.21.8. a.b.c.d.1.401	Certificates issued with additional security requirements (e.g., a smart card certificate requiring a face-to-face issuance process)
High assurance	1.3.6.1.4.1.311.21.8. a.b.c.d.1.402	Certificates issued with the utmost security requirements (e.g., a key recovery agent certificate that might require additional security background checks as part of the issuance process)

Figure 6 shows the effect of setting issuance policies in an end-entity certificate, in this case the low assurance issuance policy (which is inherited from the issuing CA at Level 1). When the certificate is used in a PKI-enabled application requiring a high assurance issuance policy, the certificate will be rejected; it can be used only in PKI-enabled applications that require a low assurance issuance policy.

**Figure 6**

*Issuance policy constraint example*



**Application policy constraints.** An application policy constraint limits the applications for which a certificate can be used. You can set an application policy in both CA (hierarchical and cross-certified) and end-entity certificates. Like issuance policies, application policies are identified by using the OID of the corresponding policy. These policies are kept in a certificate's *Application policies* extension. Table 3 lists the Windows 2003 PKI predefined application policies and their corresponding OIDs.

**Table 3 Predefined application policy constraints and corresponding OIDs**

<b>Application Policy Name</b>	<b>Corresponding OID</b>
All Application Policies	1.3.6.1.4.1.311.10.12.1
Certificate Request Agent	1.3.6.1.4.1.311.20.2.1
Client Authentication	1.3.6.1.5.5.7.3.2
Code Signing	1.3.6.1.5.5.7.3.3
Digital Rights	1.3.6.1.4.1.311.10.5.1
Directory Service Email Replication	1.3.6.1.4.1.311.21.19
Document Signing	1.3.6.1.4.1.311.10.3.12
Embedded Windows System Component Verification	1.3.6.1.4.1.311.10.3.8
Encrypting File System	1.3.6.1.4.1.311.10.3.4
File Recovery	1.3.6.1.4.1.311.10.3.4.1
IP Security End System	1.3.6.1.5.5.7.3.5
IP Security IKE Intermediate	1.3.6.1.5.5.8.2.2.1
P Security Tunnel Termination	1.3.6.1.5.5.7.3.6
IP Security User	1.3.6.1.5.5.7.3.7
Key Pack Licenses	1.3.6.1.4.1.311.10.6.1
Key Recovery	1.3.6.1.4.1.311.10.3.11
Key Recovery Agent	1.3.6.1.4.1.311.21.6
License Server Verification	1.3.6.1.4.1.311.10.6.2
Lifetime Signing	1.3.6.1.4.1.311.10.3.13
Microsoft Time Stamping	1.3.6.1.4.1.311.10.3.2
Microsoft Trust List Signing	1.3.6.1.4.1.311.10.3.1
OEM Windows System Component Verification	1.3.6.1.4.1.311.10.3.7
Private Key Archival	1.3.6.1.4.1.311.21.5
Qualified Subordination	1.3.6.1.4.1.311.10.3.10
Root List Signer	1.3.6.1.4.1.311.10.3.9
Secure Email	1.3.6.1.5.5.7.3.4
Server Authentication	1.3.6.1.5.5.7.3.1
Smart Card Logon	1.3.6.1.4.1.311.20.2.2
Time Stamping	1.3.6.1.5.5.7.3.8
Windows Hardware Driver Verification	1.3.6.1.4.1.311.10.3.5
Windows System Component Verification	1.3.6.1.4.1.311.10.3.6

In Version 2 certificates, which Windows 2003 introduced, application policies have the same function as the Win2K extended key usage (EKU) certificate extension. Version 2 certificates are generated by an enterprise CA based on a Version 2 certificate template. For downlevel compatibility, Windows 2003 CAs and Windows 2003 and XP clients can still work with the EKU extension.

As I mentioned, you can set application policies in both end-entity certificates and CA certificates. If you set an application policy in an end-entity certificate, you limit the applications for which the certificate can be used. If you set the policy in a CA certificate, the policy will be copied in all certificates (end-entity and CA) the CA issues and will thus limit the applications for which those certificates can be used. Setting the policy in a CA certificate will also limit the certificate types a CA can issue. For an enterprise CA, the application policy settings even overrule the certificate templates

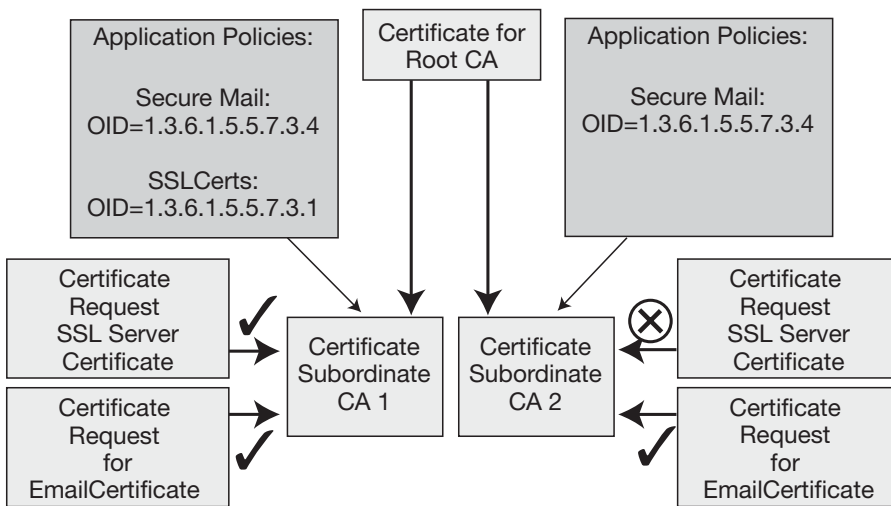
that are loaded in its Certificate Templates container. For example, if you want a subordinate CA to issue user certificates, you need to make sure that you add the application policy OIDs for the Encrypting File System (EFS), Secure Email, and Client Authentication. The User certificate template covers all three application policies.

Application policies that are set in cross-certification certificates limit the applications for which a certificate with the cross-certificate in its certificate chain can be used. In this case, enforcement of the application policy is the certificate chain validation software’s responsibility. Again, the code needed to validate the application policy is available only in Windows 2003 and XP.

Figure 7 shows the effect of setting application policies in CA certificates. The figure shows that an application policy has been set in the certificates of subordinate CA 1 and CA 2. Subordinate CA 1 will accept both email and Secure Sockets Layer (SSL) certificate requests. Subordinate CA 2 can issue only email certificates and will reject SSL certificate requests.

**Figure 7**

*Application constraint policy example*



## Defining Trust Constraints

Windows 2003 PKI offers three tools to define PKI trust constraints: the capolicy.inf configuration file, the policy.inf configuration file, and the Microsoft Management Console (MMC) Certificate Templates snap-in.

During CA installation, you can use the capolicy.inf configuration file to set a CA certificate’s PKI trust constraints. You can also use the configuration file to define other CA configuration settings, such as certificate revocation list (CRL) Distribution Points and Authority Information Access (AIA) locations. The content of the capolicy.inf file is checked for trust constraints at CA installation and every time the CA certificate is renewed. You need to store the file in the %systemroot% folder of the machine on which the CA is installed, and you can’t change the file’s name. You can use the capolicy.inf file to define only basic and issuance policy constraints.

The `policy.inf` configuration file defines the PKI trust constraints that are embedded in a CA certificate request file, and the `Certreq` utility uses this file as a parameter. `Policy.inf` is the most complete trust constraint configuration tool; contrary to the `capolicy.inf` file, you can use it to configure all the different categories of PKI trust constraints. As opposed to the `capolicy.inf` file, you can change the name of the `policy.inf` file.

You can use the Certificate Templates snap-in to create, modify, or delete certificate templates. Certificate templates define the properties (including the PKI trust constraints) of certificates issued by Windows CAs. You can modify the content of Version 2 certificate templates; you can't modify Version 1 certificate templates. Certificate templates don't offer the same level of granularity for PKI trust-constraint definition as is possible with a `policy.inf` configuration file: You can use templates to set only basic, application policy, and issuance policy constraints.

## Flexible PKI Trust Definition

Trust is a fundamental PKI concept. Windows 2003 PKI's enhanced trust features make Windows PKI more powerful and flexible but also add more complexity to PKI trust design and administration. Still, no other security protocol or technology available today can define trust in such a granular way.

## Chapter 3:

# User-Side PKI Trust Management

—by *Jan De Clercq*

One of the most important concepts in a public key infrastructure (PKI) is trust: PKI administrators and users must be able to determine which public keys are trustworthy. In Chapter 2, “CA Trust Relationships in Windows Server 2003 PKI,” I discussed the primary Windows 2003 PKI trust models—hierarchical and networked—and explain the concept of constrained trust in Windows 2003 PKI. These topics are primarily about Certification Authorities (CAs) and servers in a PKI trust.

However, if you want to establish a reliable PKI, you also need to understand how PKI administrators manage PKI-user-side trust decisions. In this context, the concept of a trust anchor (i.e., a CA that the PKI user explicitly trusts under all circumstances) is particularly important.

Windows 2003 and Windows XP include several mechanisms to control a PKI user’s trust anchors. Some are user-driven mechanisms; others are Local Machine Administrator-driven or even Domain or Enterprise Administrator-driven mechanisms. The administrator-driven mechanisms are available only when the PKI client is a member of a Windows 2003 domain and forest infrastructure. Table 1 lists the available mechanisms and their characteristics, which I discuss in more detail in the next sections.

**Table 1 User PKI trust management mechanisms**

Mechanism	Scope	Managed By	Management Interface or Mechanism
Machine certificate store	Machine	Local Administrator	MMC Certificates snap-in
User certificate store certificates viewer	User	User	MMC Certificates snap-in, IE
Enterprise Trust (CTLs)	Depends on the AD object that the GPO is linked to	GPO Administrator	GPO Editor
Trusted root CAs	Depends on the AD object that the GPO is linked to	GPO Administrator	GPO Editor, certutil.exe -dspublish RootCA command
NTAuth store	Forest	Forest or Domain Administrator	Certutil.exe -dspublish NTAuth command
Windows Update	All machines with the Root Certificate Update Service enabled	Forest or Domain Administrator, Microsoft	Microsoft Root Certificate Program

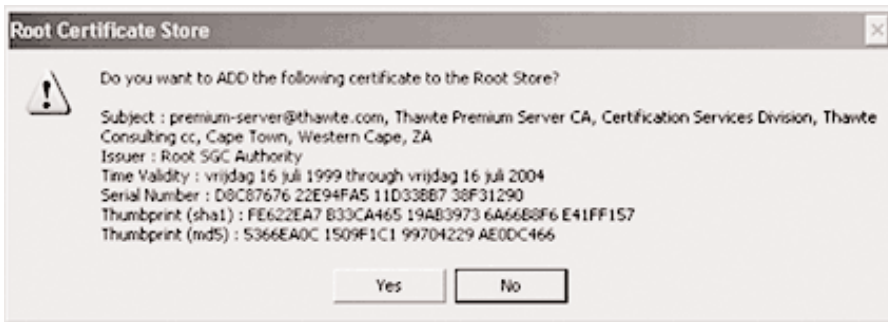
## User-Centric PKI Trust Management

Windows 2003 and XP contain functionality to let PKI users make their own trust decisions. The key to this functionality is a user's certificate store and, more specifically, the trusted root CA's certificate container (aka the root certificate store). To access your personal certificate store, you can use the Microsoft Management Console (MMC) Certificates snap-in or the Microsoft Internet Explorer (IE) certificates viewer. To open the certificates viewer, open IE, select Internet Options, go to the Content tab, and click Certificates.

All CA certificates in the root certificate store container are by default considered trust anchors, and by default, a PKI user controls which CA certificates he or she wants to add to or remove from this container. When a user tries to add a CA certificate to the root store, a dialog box opens that asks the user to confirm that he or she wants to add the certificate to the root store, which Figure 1 shows.

**Figure 1**

*Root Certificate Store dialog box*



In a default Windows 2003 or XP installation, the root certificate store comes prepopulated with a set of CA certificates so that the user doesn't need to add all CA certificates to his or her store. However, using these certificates isn't a sound security practice; the user is relying on the software vendor's judgment to decide whether a certificate is trustworthy. Enterprises should remove all prepopulated CA certificates and add only the certificates that the IT department considers trustworthy. (In consumer environments, the prepopulated root store is a good solution from an ease-of-use perspective because it removes some of the complexity of working with PKI and PKI-enabled applications.)

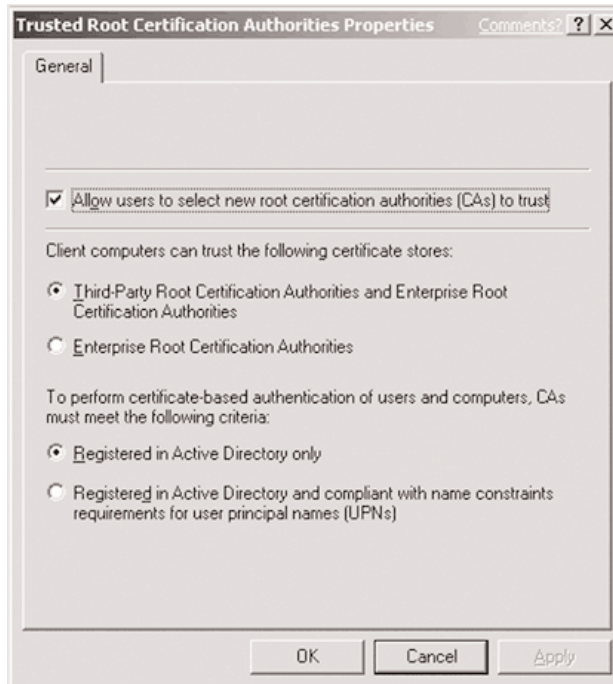
Windows 2003 comes with an important new Group Policy Object (GPO) trust management extension. The extension lets administrators set whether a user is allowed to make his or her root certificate store trust decisions and to determine which certificate store containers are considered trust anchor stores. To access the new settings, open the

MMC Group Policy Object snap-in, then open the Computer Configuration, Windows Settings, Security Settings, Public Key Policies, Trusted Root Certification Authorities GPO container, and select Properties. To let users make their own trust anchor decisions, select the *Allow users to select new root certification authorities (CAs) to trust* check box, as Figure 2 shows. If you set *Client computers can trust the following certificate stores* to Enterprise Root Certification Authorities, only the certificates stored in the CN=Certification Authorities,CN=Public Key Services,CN=Services,CN=Configuration,

DC=<domain>,DC=<domain> AD container will be trusted. If you select *Third-Party Root Certification Authorities and Enterprise Root Certification Authorities*, the certificates in the above Active Directory (AD) container and the Ones in the certificate store's Third Party Root Certification Authorities container will be trusted.

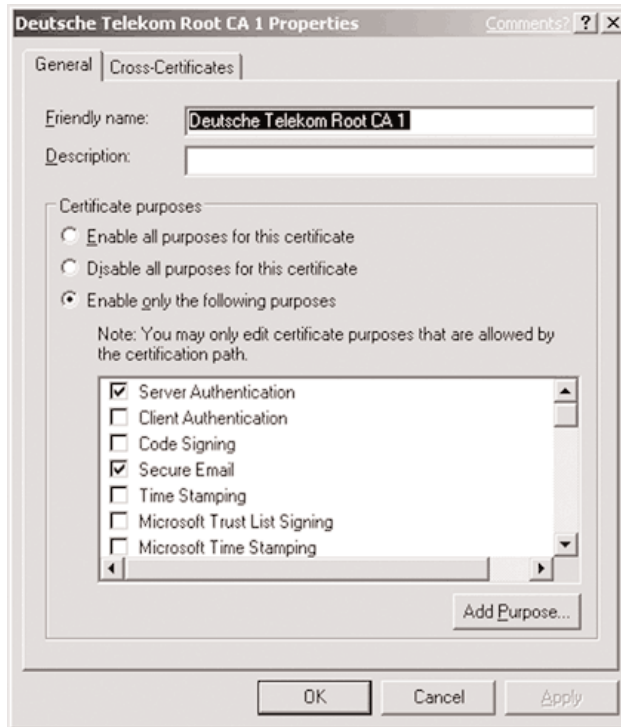
**Figure 2**

*Trusted Root Certification Authorities Properties dialog box*



Independent of the above settings, users can always set the applications or purposes for which they want to trust a particular certificate in their certificate store. To access this functionality, a user needs to open *Certificate properties* in the Certificates snap-in, go to the Details tab, click Edit Properties, select *Enable only the following purposes*, and select the applications or purposes for which he or she wants to trust the certificate, as Figure 3 shows. Setting this certificate property affects the selected applications the same as if the certificate contained an extended key usage (EKU) or Application Policy X.509 certificate extension.

**Figure 3**  
*Setting certificate trusts*



Most of the trust anchor certificates in the root store are inherited from the local machine certificate store. Only the local administrator can directly modify the trust anchors on the local machine. To view the content of a machine's certificate store, open the Certificates snap-in and select the local machine. To see the certificates in their personal certificate store that are inherited from the local machine store, users can select *Show physical certificate stores* in the View options of their personal certificate store. Each Logical Certificate container holds a Local Computer container that stores the certificates inherited from the local machine certificate store.

## Centralized User PKI Trust Management

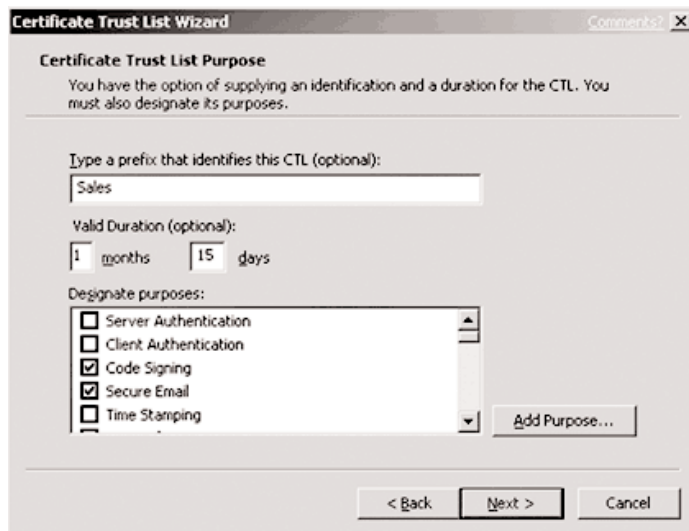
Windows 2003 provides three ways to centrally control a PKI user's trust anchors. You can manage trust anchors by using GPO settings, the NTAUTH AD store, or the Windows Update service.

The two GPOs that let you control a user's trust anchors are the Trusted Root Certification Authorities GPO and the Enterprise Trust GPO. Both GPOs are located in the Computer Configuration, Windows Settings, Security Settings, Public Key Policies GPO container. The GPO settings are automatically downloaded to PKI clients as part of the Group Policy application process on the Windows client.

The Trusted Root Certification Authorities container is used to distribute trustworthy Enterprise CA certificates to PKI users. The CA entries in this container have unlimited trust (as long as the certificates haven't expired).

The Enterprise Trust container contains a set of certificate trust lists (CTLs), which are signed lists of CA certificates. The certificates are considered trust anchors only if the CTL is signed by using a private key whose public key certificate has been issued by another trust anchor. Administrators can limit how long the CTL entries are valid and for which applications they are valid. To do so, open the Group Policy Object snap-in, navigate to the User Configuration\ Security Settings\ Public Key Policies\ Enterprise Trust container, right-click it, and select New, Certificate Trust List to open the Certificate Trust List Wizard, which Figure 4 shows.

**Figure 4**  
*Certificate Trust List Wizard*



The NTAAuth AD store is a special trust anchor store. It holds the CA certificates of all Windows 2003 Enterprise CAs and CAs that are trusted to issue Windows smart card logon certificates or certificates that contain a client authentication EKU or application policy (e.g., for use with Secure Sockets Layer—SSL—client authentication or RAS and VPN authentication). The NTAAuth trust anchor certificates are downloaded to every PKI client as part of the Windows autoenrollment event. An autoenrollment event occurs when a user logs on, when an administrator uses the Gpupdate utility to manually refresh the local GPOs, or during an automatic Group Policy refresh (which occurs every 90 minutes by default). The NTAAuth certificates are stored in the cACertificate attribute of the NTAAuth Certificates object that's in CN=Public Key Services ,CN=Services, CN=Configuration,DC=<domain>.

The third centralized user PKI trust management solution is the Root Certificate Update Service, which is a Windows Update extension. This service provides a dynamic CA certificate distribution mechanism that can replace the preloaded CA certificates. You install the required client-side software

through the Windows 2003 and XP Update Root Certificate component in the Control Panel Add/Remove Programs applet's Add/Remove Windows Components option.

The Root Certificate Update Service uses a special CTL, called the Windows Update CTL, to automatically download CA certificates when the Windows 2003 or XP client-side certificate-validation software checks the appropriate Windows Update download location. The service downloads new root CA certificates to the Third-Party Certification Authorities container in the machine and user certificate stores. Organizations that want to use this feature to distribute their CA certificate must subscribe to the Microsoft Root Certificate Program. More information about this program is available from the Microsoft TechNet site at <http://www.microsoft.com/technet/security/news/rootcert.mspx>.

## Flexible PKI Trust Definition

Trust is a fundamental concept of PKI. The enhanced trust features of Windows 2003 PKI simplify PKI user-side trust management and enable PKI users to make some trust decisions on their own. Every PKI user should have some understanding of how he or she can make basic PKI trust decisions.

## Chapter 4:

# Validating Digital Certificates in Windows PKI

—by Jan De Clercq

Certificate validation is a key part of the process of authenticating users and systems and securing network communications through the use of digital certificates. To validate a digital certificate, a Windows public key infrastructure (PKI)-enabled application must determine whether the certificate and the public key it contains are trustworthy.

Validating a certificate requires the certificate-validation logic in the PKI-enabled application to perform a series of checks on different parts of the certificate. Let's examine those checks and other aspects of the certificate-validation process. By gaining an in-depth understanding of how certificate validation works, you'll be better prepared to recognize and solve certificate-validation problems when they occur.

## Certificate-Validation Checks

The validation process performs the following checks on a certificate: digital signature, trust, time, revocation, and formatting. A certificate is invalid if it doesn't pass one or more of these checks. During the digital signature check, the validation software uses a trustworthy public key to validate the digital signature that the certificate issuer (i.e., the Certificate Authority—CA) has applied to the certificate content. The key can be the public key of the issuing CA or of another CA that's part of the certificate's *certificate chain*—a hierarchical trust model that I explain later.

The availability of a public key isn't enough to validate a signature; the public key must also be trusted. In the Windows Server 2003 and Windows 2000 Server PKI, a trusted CA certificate and public key are known as a *trust anchor* and are available from the Trusted Root Certification Authorities container in a Windows PKI client's certificate store. The trust check performs the process of authenticating a trusted CA certificate—a procedure also called certificate-chain validation. Certificate-chain validation might trigger different certificate-validation loops for each certificate in the chain. I explain certificate-chain validation in more detail later.

During the time check, the validation process compares the certificate's start and end dates with the current time. One reason a certificate's lifetime is limited is to accommodate advances in computer security, particularly in cryptography. You don't want to rely on certificates based on obsolete technology.

The revocation check determines whether the issuing CA has revoked the certificate. The PKI in Windows 2003 and Win2K Server supports complete certificate revocation lists (CRLs) and CRL Distribution Points (CDPs). In addition, Windows 2003 Certificate Services supports delta CRLs. CRLs, CDPs, and delta CRLs can provide automated certificate-revocation checking. (For more information about certificate revocation, see Chapter 6, "Understanding Windows PKI Certificate Revocation.")

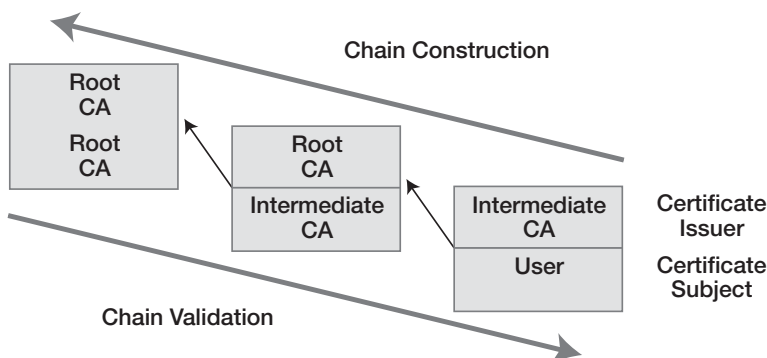
The validation process's formatting check validates the certificate's format against the standard certificate format as defined in the International Telecommunications Union Telecommunication Standardization Sector (ITU-T) X.509 recommendation. This check also validates certificate extensions, which include basic constraints, name constraints, application policy constraints, and issuance policy constraints. (For more information about these extension types, see Chapter 2, "CA Trust Relationships in Windows Server 2003 PKI.") Most Secure MIME (S/MIME) applications, for example, evaluate the certificate subject's Internet Engineering Task Force (IETF) Request for Comments (RFC) 822 name (i.e., the standard Internet mail address format—such as jan.declercq@hp.com) of a signed message and compare it with the sender entry in the SMTP message header. In the case of S/MIME, this check protects against impersonation or man-in-the-middle attacks. In such attacks, a malicious entity reuses a user's identity to gain access to a system or network. Most Secure Sockets Layer (SSL) implementations perform a similar validation check. SSL compares the certificate subject's RFC 822 name with the name contained in the URL of the secure Web site that the client is accessing.

## Regular Certificate-Chain Processing

What's a certificate chain, and why does it need to be processed during certificate validation? The certificate chain provides a way to verify that all certificates related to the certificate being validated are trustworthy. To understand certificate chains, let's look at the example of a hierarchical PKI trust model. In a hierarchical trust model (which I also discuss in Chapter 2), each end-entity's certificate chain consists of all CA certificates that form the path between the user and the root CA in the PKI hierarchy. In the hierarchical PKI trust model, each certificate contains a pointer to its parent—or issuing—CA, which is stored in the issuer field of an X.509 certificate. Figure 1 shows the certificate chain of a user certificate that a CA has issued and that's part of a two-level PKI hierarchy. Figure 1 represents certificates in a simplified way by using the certificate subject and certificate issuer. In this example, the user's certificate subject is the user, and its issuer is the intermediate CA. The intermediate CA's certificate subject is the intermediate CA, and its issuer is the root CA. In a hierarchy, the root CA always has a self-signed certificate, which means the certificate subject and issuer are identical.

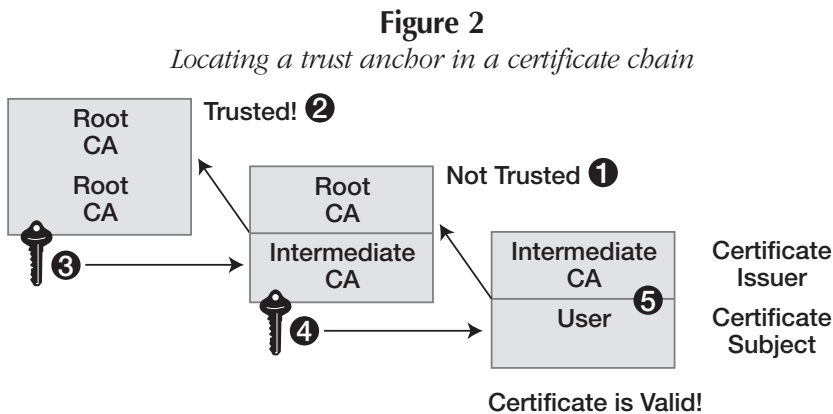
**Figure 1**

*Certificate-chain-processing overview*



The certificate-validation software processes a certificate's certificate chain. This process can be split into two subprocesses: chain construction and chain validation.

**Chain construction.** During chain construction, the certificate-validation software walks through the certificate's chain starting with the user certificate until it finds a trusted CA certificate (i.e., the trust anchor). In the example that Figure 2 shows, the validation software finds a trust anchor at the root CA level. Alternatively, the validation software could find a trust anchor at the intermediate CA level. When a trust anchor is found, the chain-construction subprocess stops and the validation logic switches to chain validation. If the certificate-validation software can't find a trust anchor, the certificate-chain process stops, preventing the validation process from making any decisions about the certificate's trustworthiness.



**Chain validation.** During chain validation, the certificate-validation software walks through the chain in the opposite direction starting with the trust anchor found by the chain-construction process and validates every CA certificate that's part of the chain. For a certificate to be validated, it must be available locally in a container in the user's certificate store. I explain what happens when a certificate isn't available locally later.

The identification of a CA certificate during chain validation is based on the Authority Key Identifier (AKI) certificate extension of the certificate being verified. A certificate's AKI field can contain different types of information:

- Issuer name and serial number of the issuer's certificate—If the AKI field contains this information, the chain-validation software tries to find a matching certificate by using the certificate's *Serial number* and Subject fields. This method of identifying a certificate is called an *exact match*.
- Public Key identifier (KeyID) of the issuer's certificate—If the AKI field contains this information, the chain-validation logic tries to find a matching certificate by using the certificate's Subject Key Identifier (SKI) extension, which contains a unique identifier for a certificate subject's public key. This method of identifying a certificate is called a *key match*.

If the certificate being verified doesn't contain an AKI field, the chain-validation software tries to identify the issuing CA's certificate by matching the name in the Issuer field of the certificate being

verified with the name in a certificate's Subject field. This method of identifying a certificate is called a *name match*.

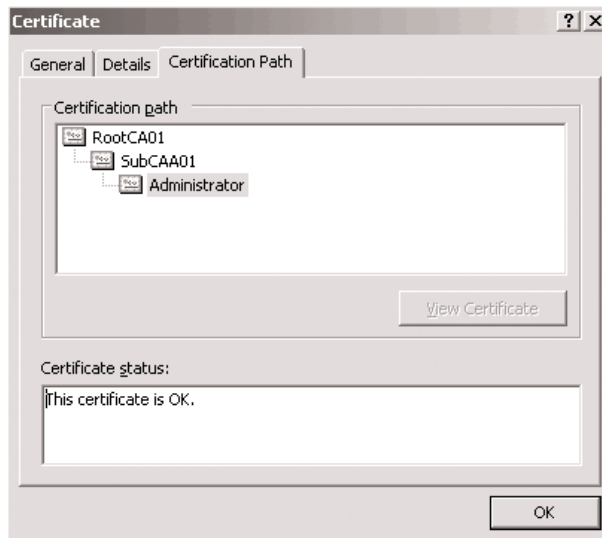
When a certificate isn't available locally, the Windows certificate-validation software uses the Authority Information Access (AIA) extension to obtain a copy of the certificate by downloading it from an online location. To do this, the validation software uses a certificate's AIA field, which contains an FTP, HTTP, Lightweight Directory Access Protocol (LDAP), or file system drive pointer to a location in which the CA's certificate is stored. If the AIA field has multiple entries, the validation software tries all entries in the order they're listed in the AIA field. The validation software caches all certificates that it downloads from an AIA location in the PKI user's profile on the local file system (specifically, in the \Documents and Settings\username\Local Settings\Temporary Internet Files folder) and in the user's certificate store.

If the certificate isn't available online or locally, certificate verification fails. When the certificate is available, the certificate-validation logic runs (for every certificate in the chain) all the checks that I discussed earlier: digital signature, trust, time, revocation, and formatting.

You can view a certificate's certificate chain by selecting the Certification Path tab in the certificate's properties dialog box, as Figure 3 shows. To obtain an overview of all your certificates, open the Microsoft Management Console (MMC) Certificates snap-in; to view a certificate's properties, double-click the certificate in the Certificates snap-in.

When you download a certificate by using the CA Web interface in Windows 2003 or Win2K Server, you can choose to download either the certificate itself or the certificate along with all certificates that are part of its certificate chain. In some cases, you might want to download the entire certificate chain—for example, on a laptop or notebook PC—so that all the certificates in the certificate chain are easily available to the validation software when you're on the road.

**Figure 3**  
*Certification Path tab*



## CTL Certificate-Chain Processing

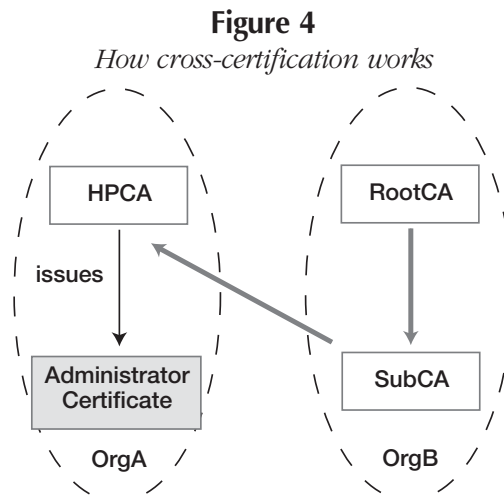
A special case of certificate-chain processing is Certificate Trust List (CTL) certificate-chain processing. A CTL is a signed list of trusted root CA certificates; that is, it can contain only self-signed root CA certificates. You define CTLs by using the pop-up menu of the Enterprise Trust Group Policy Object (GPO) container, which you can access by navigating to \Windows Settings\Security Settings\Public Key Policies. GPOs also automatically download CTLs to the Enterprise Trust container in an entity's certificate store. The Enterprise Trust container isn't a trust anchor container; by default, its content isn't considered trusted.

For a CTL and its content to be trusted, the CTL signing certificate must be valid. This means that the CTL signing certificate should pass the digital signature, time, formatting, and revocation checks. To ensure that the digital signature check succeeds, the CTL signing certificate's certificate chain should contain a certificate that's part of the Trusted Root Certification Authorities container. You can determine whether a certificate chain is part of a valid CTL by viewing each certificate in the chain on the Certificate properties' Certification Path tab, as I discussed earlier.

## Cross-Certification Chain Processing

Cross-certification is a new Windows 2003 PKI trust feature, which I explained in detail in Chapter 2. Unlike CTLs, cross-certification allows for granular PKI trust definitions between different CA entities. When you set up cross-certification between two CA entities, each CA becomes both a parent and a subordinate CA, which has interesting effects on the way certificate-chain building works.

Figure 4 shows how a cross-certified trust relationship works—and how it would appear in the certificate's properties. The CA trust relationships that are linked to this setup are on the left side of the figure. In this example, a one-way cross-certification trust exists between OrgB and OrgA. The subordinate CA—SubCA—issues a cross-certificate to HPCA (i.e., OrgA's root CA), which lets users in OrgB trust a certificate named Administrator that HPCA issued. In a nutshell, the users in OrgB trust RootCA, SubCA chains to RootCA and cross-certifies HPCA, and HPCA issues the Administrator certificate.



Certificate validation is a complex topic. The next time you have a problem with an invalid certificate, your knowledge of the basics of Windows certificate validation might help you narrow down possible causes for the problem and make solving it a little easier.

## Chapter 5:

# Windows Server 2003 PKI Certificate Autoenrollment

—by *Jan De Clercq*

Certificate autoenrollment in Windows Server 2003, Windows XP, and Windows 2000 automatically creates certificates for users and machines. Autoenrollment handles certificate enrollment, certificate renewal, and certain housekeeping tasks, such as removing revoked certificates from a user's or machine's certificate store and downloading trusted root Certification Authority (CA) certificates and cross-certificates (a new way to set up CA trust relationships in Windows 2003) from Active Directory (AD). Win2K public key infrastructure (PKI) supports certificate autoenrollment only for machine certificates and Encrypting File System (EFS) user certificates. Fortunately, Windows 2003 PKI extends certificate autoenrollment for users to all certificate types.

Windows PKI uses certificate autoenrollment several ways:

- Every Windows 2003 and Win2K domain controller (DC) automatically receives a DC certificate when the machine joins a domain in which an enterprise CA is defined.
- An administrator can set a Group Policy Object (GPO) setting that automatically enrolls machines for IP Security (IPSec) or Secure Sockets Layer (SSL) certificates.
- An administrator can set a GPO setting that automatically enrolls several users for a user or secure-mail certificate.
- A CA administrator who wants to change a property of a particular certificate type can duplicate the old certificate template to create a new certificate template and let the new template supersede the old one. Autoenrollment then automatically distributes to the appropriate PKI users a new certificate based on the new template.
- An administrator can automate the creation of certificates for new users.

Let's look at how to set up user and machine certificate autoenrollment in a Windows 2003 PKI environment. Let's also look at some of autoenrollment's nuts and bolts.

## How Autoenrollment Works

The Winlogon process triggers certificate autoenrollment (i.e., the autoenrollment event). The Winlogon process is initiated every time a user performs an interactive logon and every time an administrator applies machine- or user-based group policies. By default, the process applies group policies every 90 minutes. You can trigger GPO updates manually, and unlocking a workstation doesn't trigger a certificate autoenrollment event.

During an autoenrollment event, the client OS queries AD to download the content of a set of predefined certificate stores to the local store on the client machine. These stores include NTAAuth, the trusted root CA, certificate templates, and Authority Information Access (AIA—for cross-certificates) AD containers. Both the NTAAuth and trusted root CA containers download trustworthy CA certificates to Windows domain clients. The certificate templates container contains a definition of all certificate types that the forest supports. The AIA container lets enterprise PKI users download trusted CA certificates from AD. Autoenrollment then processes the certificate templates, analyzes their properties, and creates a requirements list of tasks to be performed during the autoenrollment event. The requirements list includes the following:

- Certificate enrollment tasks—Autoenrollment adds all templates that have autoenroll and read permissions set for the current machine or user.
- Certificate renewal tasks—Autoenrollment processes the user's or machine's MY certificate store container to look for expired certificates or certificates that are about to expire and adds these certificates to the requirements list. Automatic certificate renewal starts when 80 percent of the certificate's lifetime has passed or when the renewal interval period specified in the certificate template has been reached. The latter is specified on the General tab of a Version 2 certificate template in Windows 2003.
- Certificate enrollment tasks based on template supersede rules—Autoenrollment evaluates certificate template supersede rules and makes the appropriate additions and deletions to the requirements list. Therefore, if a new certificate template superseded a particular template, the process adds an autoenrollment task for the newer template to the requirements list.

The autoenrollment process then searches AD for an enterprise CA that can issue the certificates. If it finds a CA, it passes the requirements list to the CA, which processes the certificate-enrollment and renewal requests. If the CA issues a certificate, the autoenrollment process installs it in the user's or machine's MY certificate store container. If the certificate's state is set to pending (for certificate requests that require administrator approval), the autoenrollment process saves the request information in the user's or machine's certificate enrollment request store.

At the end of the autoenrollment process, the outcome (success or failure) of the process is logged in the local system's Application event log. If autoenrollment fails, a summary dialog box appears.

You can configure the autoenrollment process to log more verbose information and events in the Application event log. Simply set the AEEventLogLevel registry subkey (of type REG\_DWORD) to a value of 0 in the following registry subkeys:

- HKEY\_CURRENT\_USER\Software\Microsoft\Cryptography\AutoEnrollment (for user autoenrollment)
- HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Cryptography\AutoEnrollment (for machine autoenrollment)

Autoenrollment events trigger more than just certificate autoenrollment. They also download trusted root CA certificates from the AD-based CAs and NTAAuth stores to the local machine's Trusted Root Certification Authorities certificate store. The autoenrollment process doesn't download the

complete NTAAuth store, however; it downloads only the differences in the content between the user certificate and the NTAAuth store.

Autoenrollment events download cross-certificates from AD to the local machine's certificate store. As with trusted root CA certificates, the process downloads only the changes. Autoenrollment also enumerates the pending certificate requests in the user's certificate enrollment request store. After the CA issues the certificate, the process downloads the certificate and installs it in the user's certificate store. If the request has been pending for more than 60 days, the process removes the request from the user's request store.

The autoenrollment process also deletes expired and revoked certificates in the user certificate attribute of the user's AD object and in the user's local machine certificate store. The latter occurs only if you select the *Delete revoked or expired certificates* property on the Request Handling tab in the certificate template properties.

## Setting Up Certificate Autoenrollment

As with Win2K, Windows 2003 lets you enable machine-certificate autoenrollment from the GPO public key policy's Automatic Certificate Request Settings container. To start the Automatic Certificate Request Setup Wizard, which Figure 1 shows, right-click the container and select New, Automatic Certificate Request. This container is in the User Configuration, Windows Settings, Security Settings, Public Key Policies GPO container.

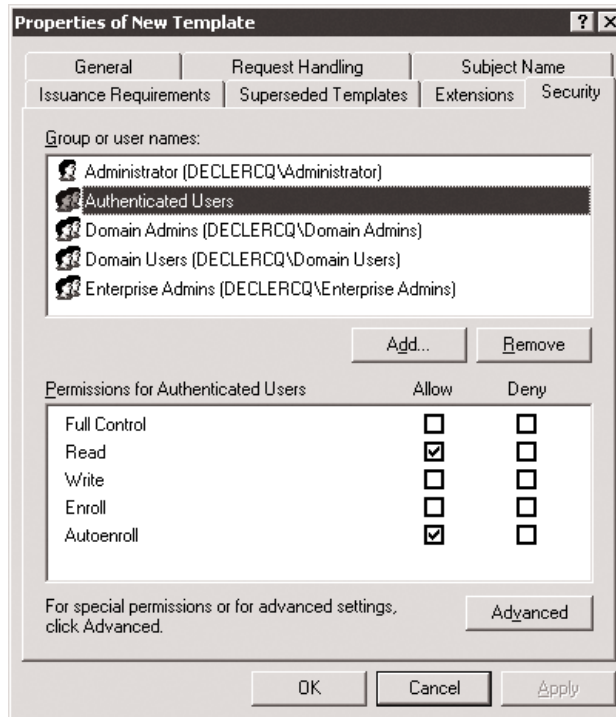
**Figure 1**  
*Automatic Certificate Request Setup Wizard*



To set up user certificate autoenrollment, you need to make configuration changes in the Microsoft Management Console (MMC) Certificate Templates and Group Policy snap-ins. To enable autoenrollment at the template level, open the Certificate Templates snap-in, then open the template.

Select the Security tab, and set the appropriate ACL settings to give users or groups autoenroll permissions, as Figure 2 shows.

**Figure 2**  
*Properties of New Template Security tab*



You can set these user autoenrollment properties only on Version 2 certificate templates. (Version 1 certificate templates shipped with Win2K; unlike Windows 2003's Version 2 certificate templates, Version 1 certificate template properties can't be changed.) Be aware that only domains with Windows 2003 schemas support Version 2 templates and only Windows 2003, Enterprise Edition or Windows 2003, Datacenter Edition AD-integrated CAs can issue certificates based on Version 2 certificate templates.

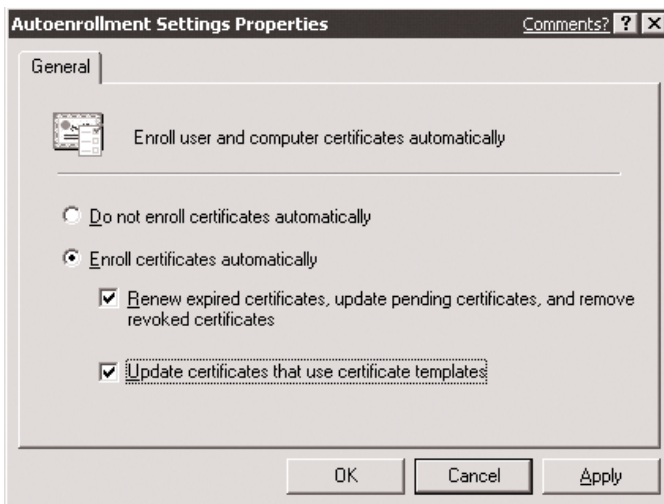
Requiring user input on machine certificate templates will make machine autoenrollment fail. To make user certificate autoenrollment occur without user intervention, leave the default settings on the Request Handling tab unchanged. If you want to prompt the user to start the autoenrollment process, select *Prompt the user during enrollment*. User input is required when you enroll smart card certificates. The user must enter a smart card in the smart card reader and, in most cases, a PIN.

To enable autoenrollment at the GPO level, open the Group Policy snap-in. Select Computer Configuration, Windows Settings, Security Settings, Public Key Policies, then open the Autoenrollment Settings Properties dialog box, which Figure 3 shows. Select the *Enroll certificates automatically* and *Update certificates that use certificate templates* check boxes. If you want the autoenrollment process

to perform certificate renewal and other certificate housekeeping tasks, make sure that you also select the *Renew expired certificates, update pending certificates, and remove revoked certificates* check box.

**Figure 3**

*Autoenrollment Settings Properties dialog box*



If you set up user autoenrollment to occur without user input, everything happens automatically without user intervention. If you set it up to occur with user input, a warning balloon appears in the user's taskbar tray. After about 15 seconds, a certificate icon replaces the warning balloon. When the user clicks the balloon or the certificate icon, a dialog box appears that prompts the user to choose whether to start the autoenrollment process. If the user clicks Remind Me Later in this dialog box, the warning balloon reappears at the next GPO refresh interval or at the next interactive logon. The warning balloon appears 60 seconds after the interactive logon sequence. If you want the balloon to appear immediately after the interactive logon sequence, set the HKEY\_CURRENT\_USER\Software\Microsoft\Cryptography\AutoEnrollment\AEExpress registry subkey (of type REG\_DWORD) to a value of 1.

## Forcing Automatic Enrollment and Renewal

You can force certificate enrollment to occur immediately, without waiting for the next logon or automatic GPO refresh. To force certificate enrollment for user and machine certificates, use the `gpupdate.exe` command-line utility to manually force a GPO update, which in turn triggers an autoenrollment event.

To force certificate enrollment only for user certificates, open the MMC Certificates snap-in and your personal certificate store. In the context menu, right-click Certificates - Current User and select All Tasks, Automatically Enroll Certificates.

You can also force renewal of specific user or machine certificate types. In the Certificate Templates snap-in, right-click the template and select Reenroll All Certificate Holders. Selecting this option increases the certificate template's version number, which triggers the autoenrollment event. To manually force a download of the root CA and of the cross-certificates that are stored in AD and are

downloaded as part of the autoenrollment process, you must delete the HKEY\_LOCAL\_MACHINE \SOFTWARE\Microsoft\Cryptography\AutoEnrollment\AEDirectoryCache registry subkey and all subordinate subkeys on the client machine.

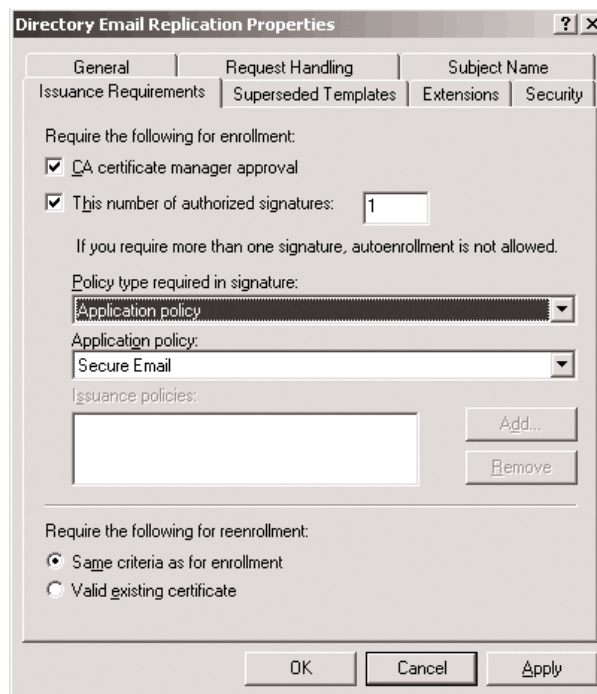
## Advanced Autoenrollment Options

Now let's look at some of the advanced autoenrollment options, such as the requirement for certificate manager approval, the selfRA feature, the concept of superseding certificate templates, and the meaning of the *Do not automatically reenroll if a duplicate certificate exists in Active Directory* certificate template property. These options are available only on Version 2 certificate templates.

Version 2 certificate templates have a property called *CA certificate manager approval* on the Issuance Requirements tab, as Figure 4 shows. If you set this property, CA manager approval is required before the CA will issue the certificate. Until the CA manager approves the request, it adds the request to the CA's pending request store. The autoenrollment process then periodically checks the CA for approved requests and automatically installs the certificates on the client machine. The CA manager can approve pending certificate requests from the pending request container in the CA snap-in.

**Figure 4**

*Directory Email Replication Properties Issuance Requirements tab*



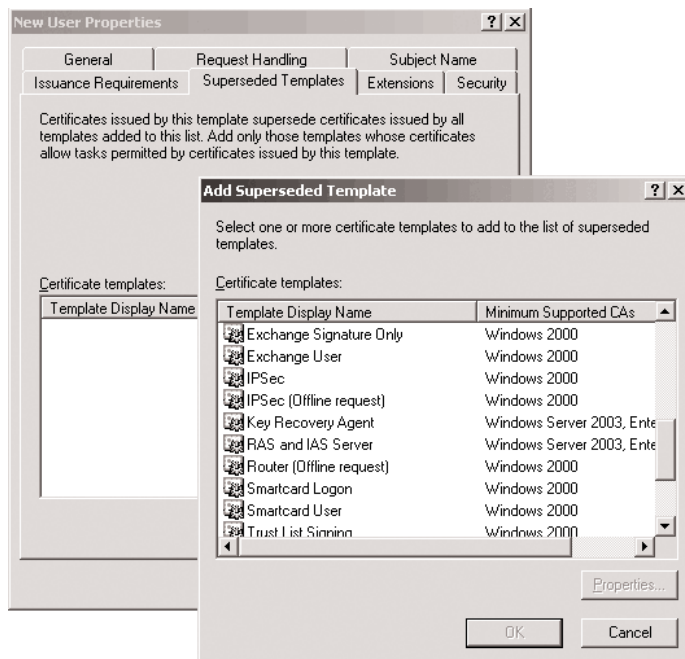
SelfRA is a new Windows 2003 PKI feature that lets you set special enrollment requirements on Version 2 certificate templates. To sign a new certificate request, selfRA requires an existing

(previously issued) certificate and its associated private key. SelfRA is also configured from the Issuance Requirements tab on the certificate template properties and works in conjunction with autoenrollment. However, autoenrollment can't deal with requests that require more than one signature to authorize the enrollment request. You can set the following selfRA-related properties:

- The number of signatures required to authorize the certificate request (for autoenrollment, the number of signatures is limited to one)—This property might be required when you issue certificates for applications with high security requirements; in that case, you might want to make certificate issuance dependent on the approval of various entities.
- The content of the application and issuance policy fields in the authorization certificate's X.509 extensions.
- The requirements for automatic reenrollment—Use the same criteria you used for the original enrollment (listed in the upper part of the Issuance Requirements tab) or check to determine whether a valid certificate of the type mentioned in the certificate template is present in the PKI user's certificate store.

Superseding certificate templates let CA administrators automatically reenroll users for certain certificate types. For example, you can change a property of a particular certificate type (e.g., the lifetime or content of an X.509 extension) by issuing a new certificate. To set up superseding templates, click Add on the Superseded Templates tab of the New User Properties dialog box of a Version 2 certificate template. Figure 5 shows the Add Superseded Template dialog box.

**Figure 5**  
*Add Superseded Template dialog box*



*Do not automatically reenroll if a duplicate certificate exists in Active Directory* is another useful autoenrollment certificate template property that's available under the General tab of a Version 2 certificate template. When you enable this property, autoenrollment won't enroll a user for a certificate if a similar certificate exists in the user's AD object, even if a certificate doesn't exist in My Container in the user's certificate store. The autoenrollment process queries AD to determine whether to enroll the user. This option is useful for users who don't have roaming profiles and who log on to multiple machines. Without this setting, those users would be automatically enrolled for a certificate on every machine they log on to.

## Ease of Use

Certificate autoenrollment is a useful feature from a PKI user's point of view. Compared with the feature set of other PKI products on the market, Windows 2003 PKI autoenrollment is a unique feature that gives Windows 2003 an important advantage.

## Chapter 6:

# Understanding Windows PKI Certificate Revocation

—by *Jan De Clercq*

One of the most important aspects in the design of a public key infrastructure (PKI) is certificate revocation or, more specifically, automated revocation checking. Certificate revocation ensures that the PKI system adds a certificate's serial number to a blacklist, called the certificate revocation list (CRL), when a PKI user's private key is compromised. Certificate revocation also guarantees that the PKI system efficiently distributes the revocation information to all PKI clients and PKI-enabled applications. If your PKI systems need to handle confidential or valuable information or transactions, you'll need to understand the process of revoking a certificate, Windows PKI-enabled application revocation checking support, and automated revocation-checking solutions. Let's begin by taking a closer look at CRLs.

## Certificate Revocation Lists

The International Telecommunications Union Telecommunication Standardization Sector (ITU-T) X.509 standard and Internet Engineering Task Force (IETF) Request for Comments (RFC) 2459 define a CRL, which contains a timestamped list of revoked certificates that the Certification Authority (CA) signs and makes available to PKI users in a public repository. A CRL identifies each revoked certificate by its certificate serial number. The X.509 standard defines two primary types of CRLs: complete CRLs and delta CRLs.

**Complete CRLs.** In their most basic form, CRLs are known as complete CRLs (aka base CRLs or full CRLs). Complete CRLs tend to be huge because the revocation information accumulates over time. Although Windows CRLs support versioning, each new CRL version automatically inherits all revocation information from the preceding version. So, a CRL will grow in size until certificates start expiring. Also, with each new CRL version, the client must download the complete CRL, which isn't an efficient use of network bandwidth. As a result, many administrators configure longer CRL lifetimes to reduce the number of CRL versions. But long CRL lifetimes reduce the revocation information's timeliness because new revocation information isn't immediately available.

In Windows Server 2003, you can use delta CRLs, which I explain below, to get around the complete CRL deficiencies. To limit the size of complete CRLs in a Windows 2000 PKI environment, you can do one of three things:

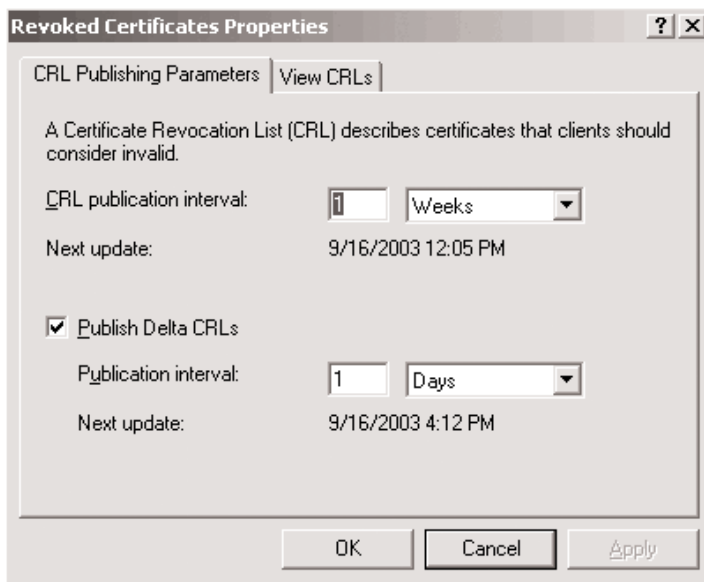
- Define multiple CAs—If you define multiple CAs and each CA maintains its own CRL, the size of individual CRLs will be much smaller than the size of one CRL that one CA generates.
- Generate certificates with a short lifetime—Win2K CRLs are self-cleaning, which means that the CA automatically removes expired certificates from the CRL.

- Generate a new CA key pair—Every time the CA renews the key pair, it generates a new CRL. The CA will use the newly generated private key to sign the new CRL.

Both Windows 2003 and Win2K publish CRLs at regular intervals. With both OSs, a CA administrator can also force the publication of a new CRL. To configure complete CRL publication intervals, open the Microsoft Management Console (MMC) Certification Authority snap-in, right-click the Revoked Certificates container, and select Properties from the menu to display the Revoked Certificates container's Properties dialog box, which Figure 1 shows. You can force the publication of the CRL by right-clicking the Revoked Certificates container in the Certification Authority snap-in and selecting the All Tasks\Publish menu option. This action opens the Publish CRL dialog box, which will ask you to specify which type of CRL you want to manually publish: a new CRL (i.e., a complete CRL) or a delta CRL.

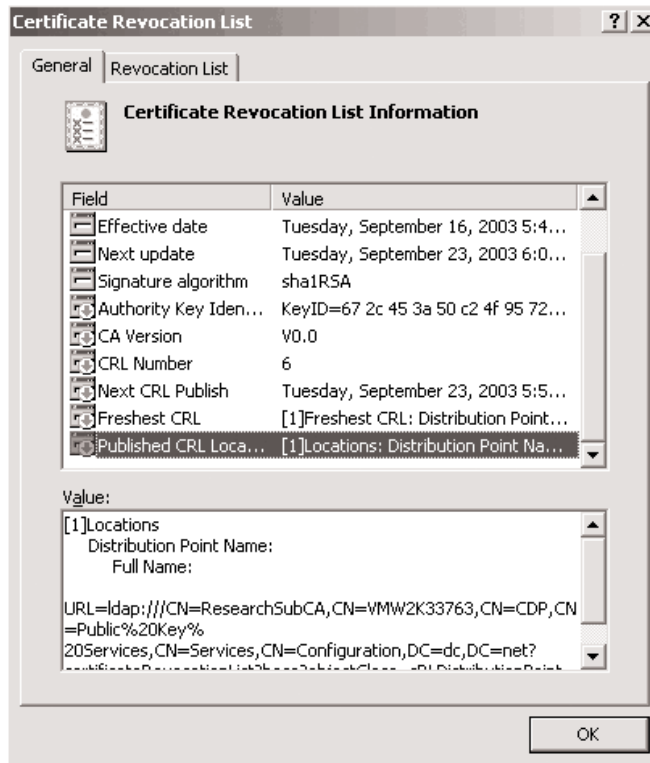
**Figure 1**

*Revoked Certificates Properties dialog box*



To view a CRL's contents and format, select the View CRLs tab in the Revoked Certificates container's Properties dialog box. When you click View CRL or View Delta CRL from the View CRLs tab, you'll see the built-in CRL viewer, which Figure 2 shows. The General tab shows the layout of a complete CRL that a Windows 2003 or Win2K CA issued. Notice the presence of some typical CRL extensions, including *Effective date*, *Next update*, CA Version, CRL Number, Next CRL Publish, Freshest CRL, and Published CRL Locations. Click the Revocation List tab in the same dialog box to view a list of the revoked certificates on a CRL.

**Figure 2**  
*The CRL viewer*



**Delta CRLs.** Windows 2003 resolves bandwidth and revocation information timeliness problems by introducing delta CRLs. RFC 3280 and RFC 2459 define delta CRLs, which are relatively small CRLs that contain only revocation changes made since the CA created the most recent complete CRL. Because delta CRLs are small, PKI clients can download them on a more frequent basis than complete CRLs, and the CA can provide more accurate revocation information to its clients. Only Windows XP Professional Edition and later Windows clients can check a certificate's validity against a delta CRL.

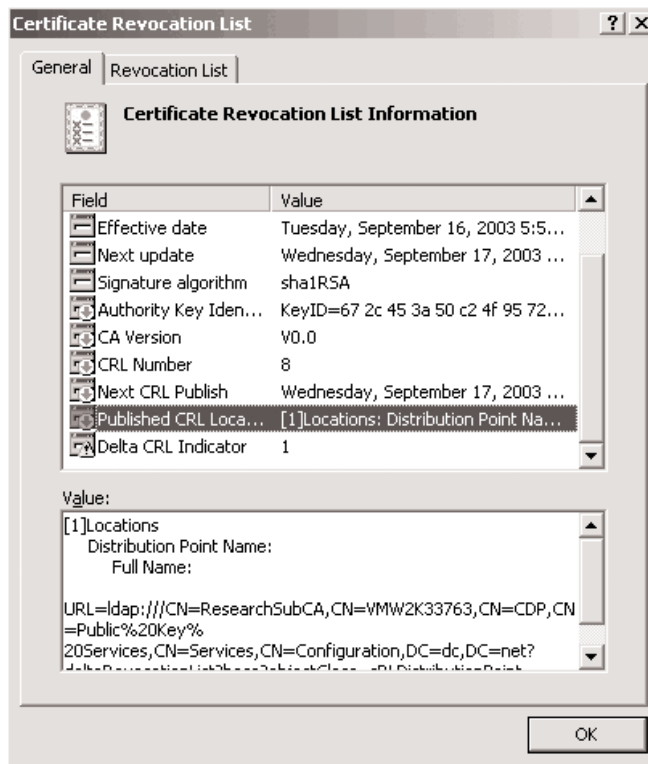
As with complete CRLs, Windows clients cache delta CRLs. If a complete CRL expires, the client retrieves a new complete CRL from the CRL Distribution Point (CDP) specified in the certificate (more on CDPs later). If the complete CRL is valid but the cached delta CRL is expired, a Windows client retrieves only the delta CRL from the CDP mentioned in the certificate.

Similar to the steps you take to manage complete CRLs, you configure delta CRL settings and view delta CRLs' content and formatting in the CA's Revoked Certificates container's Properties dialog box, which Figure 1 shows. Likewise, you follow the same procedure to manually force delta CRL publication as you use to manually force complete CRL publication.

Figure 3 shows the layout of a delta CRL that a Windows 2003 CA issued as it appears in the built-in CRL viewer. Notice the presence of the Delta CRL Indicator extension, which shows that this

CRL is a delta CRL, not a complete CRL. The value in the Delta CRL Indicator extension is the number of the complete CRL that the delta CRL must be associated with. As with a complete CRL, a list of the revoked certificates on a delta CRL is available from the Revocation List tab.

**Figure 3**  
*A delta CRL layout*

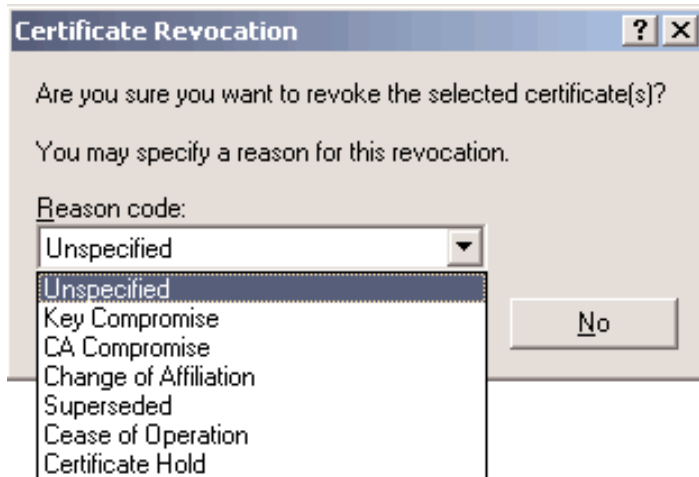


## Revoking a Certificate

A Windows CA administrator can revoke a certificate from the Certification Authority snap-in or from the command line. From the Certification Authority snap-in, open the Issued Certificates container, right-click the certificate you want to revoke, then select the All Tasks\Revoke Certificate menu option. To revoke a certificate from the command line, type the following command on the machine hosting your CA:

```
certutil revoke <certificate
serial number> <reason code>
```

When revoking a certificate, the CA administrator can specify a revocation reason code, as Figure 4 shows. Valid revocation reason codes are Unspecified, Key Compromise, CA Compromise, *Change of Affiliation*, Superseded, *Cease of Operation*, and Certificate Hold.

**Figure 4***Specifying a revocation reason code*

## PKI-Enabled Application Revocation Checking Support

Not all Windows PKI-enabled applications automatically perform revocation checking. Also, revocation checking is sometimes dependent on an application-specific configuration setting. Table 1 provides an overview of how the most commonly used Windows PKI-enabled applications support revocation checking.

**Table 1 PKI-enabled application revocation checking support**

PKI-Enabled Application Name	Revocation Checking Support
Encrypting File System (EFS)	Win2K EFS doesn't support revocation checking. Windows 2003 and XP EFS support revocation checking when you add other users to the EFS settings of a file set up for EFS file sharing.
Microsoft Internet Explorer (IE)—Authenticode code signing	Revocation checking is a configuration option on the Advanced tab of Internet Options: Check for publisher's certificate revocation.
IPSec	Win2K doesn't support revocation checking. You can edit the registry to enable revocation checking in Win2K SP2 and later by using the values outlined in the article.
Microsoft IIS—Secure Sockets Layer (SSL)—Transport Layer Security (TLS)	Internet Information Services (IIS) 5.0 and later enable revocation checking by default.
IE—SSL-TLS	Revocation checking is a configuration option on the Advanced tab of Internet Options: Check for server certificate revocation (requires restart).
Outlook—S/MIME	Revocation checking is enabled by default in Office Outlook 2003 and Outlook 2002. You can also edit the registry to enable revocation checking in Outlook 2000 SR1 by using the values outlined in the article.
Smart Card Logon	The Smart Card Logon authentication logic enables revocation checking by default.

To enable revocation checking for Secure MIME (S/MIME) in Microsoft Outlook 2000 Service Release 1 (SR1), open the registry on the client, navigate to the HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Cryptography{7801ebd0-cf4b-11d0-851f-0060979387ea} registry subkey, create a new entry called PolicyFlags (of type REG\_DWORD), and set its value to 00010000. To enable revocation checking for IP Security (IPSec) in Win2K Service Pack 2 (SP2) and later, open the registry, navigate to the HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\PolicyAgent\Oakley registry subkey, create a new entry called StrongCrlCheck (of type REG\_DWORD), and set its value to 1 or 2. This registry entry's values have the following meanings:

- 0—Disables CRL checking for certificate-based IPSec authentication.
- 1—Enables CRL checking and fails the certificate-validation process only if the CRL explicitly shows that the certificate is revoked. The client system will ignore all other failures, including when the CDP URL is unavailable.
- 2—Enables CRL checking and fails certificate validation on any CRL check errors.

## Automated Revocation Checking

In the PKI world, different models are available for automated revocation checking. Most, with the exception of certificate revocation trees and the Online Certificate Status Protocol (OCSP), are based on complete CRLs, authority revocation lists, CDPs, enhanced CRLs, delta CRLs, and indirect CRLs. I don't have room to discuss all these methods, but for a good overview of automated revocation-checking methods, see Carlisle Adams and Steve Lloyd, *Understanding Public-key Infrastructure: Concepts, Standards, and Deployment Considerations* (Que Publishing, 1999).

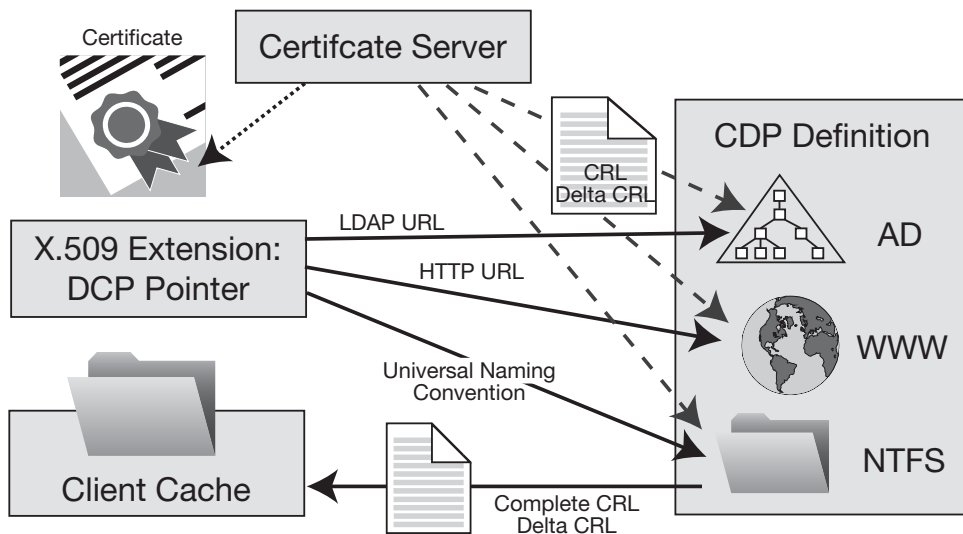
As I mentioned earlier, Win2K PKI supports complete CRLs and CDPs. Windows 2003 PKI adds support for delta CRLs. Windows 2003 and Win2K also support specific Netscape revocation extensions.

A revocation requirement often mentioned in the IT industry is support for OCSP. As RFC 2560 defines, OCSP offers real-time certificate revocation information to all PKI users. Neither Windows 2003 nor Win2K supports OCSP out of the box. However, you can add support by using third-party software from vendors such as Alacris or Valicert (a division of Tumbleweed Communications). CDPs and delta CRLs offer a native Windows alternative to using OCSP.

## CRL Distribution Points

CDPs offer a convenient way to automate revocation checking. Each certificate that a Windows 2003 or Win2K CA generates can include one or more CDPs. The CA stores CDPs in an X.509 certificate extension that it generates. A CDP can be an HTTP URL, a Lightweight Directory Access Protocol (LDAP) URL, or a file share. After the CA issues a certificate, you can't modify that certificate's CDPs. Figure 5 illustrates how CDPs work. The Certificate Server issues the certificate (shown in the figure connected by a dotted line) to the PKI clients and issues the CRLs (shown connected by a dashed line) to the CDPs. The certificate's CDPs point to the different CDP locations from which the client can download the CRLs.

**Figure 5**  
*How CDPs work*



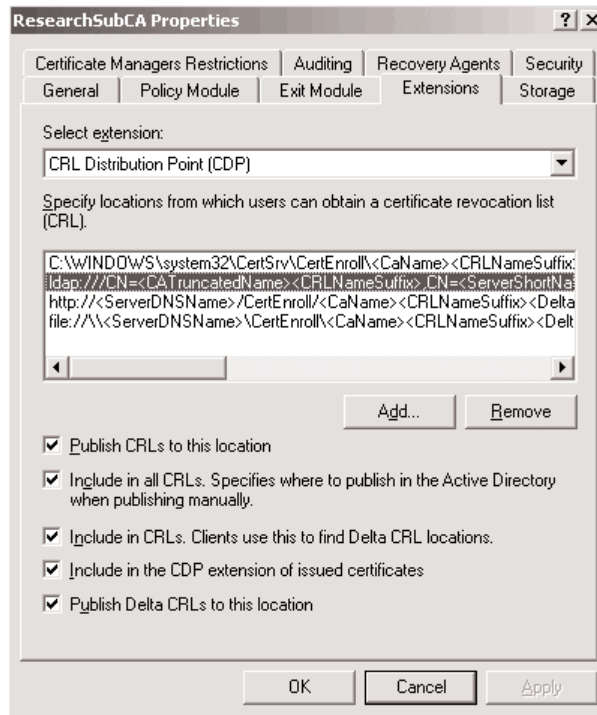
A Windows PKI-enabled application that doesn't find a local copy of the complete CRL or delta CRL will check the certificate's CDPs for an up-to-date complete CRL or delta CRL. If a complete CRL or delta CRL is available from the CDPs, the application will download the CRL and cache it locally for the lifetime of the CRL. If the certificate doesn't contain any CDPs, the application will query the certificate's issuing CA for a complete CRL or delta CRL.

For CDPs to function correctly, not only is certificate and PKI-enabled application support required, but the CA must also support them. The CA must have an exit module that can publish the complete CRLs and delta CRLs to the appropriate file system, Web, or Active Directory (AD) CDP. By default, every Windows 2003 and Win2K CA includes an exit module that can handle CRL publication. Neither Windows 2003 nor Win2K can automatically publish complete CRLs or delta CRLs to HTTP CDPs—however, you can publish these types of CRLs manually.

Besides automated revocation checking, CDPs also can increase complete CRL and delta CRL availability. Each certificate can contain multiple CDPs so that if one CDP is unavailable, the PKI logic will try another CDP.

To configure the contents of a certificate's CDP fields, open the Certification Authority snap-in, right-click a Windows CA object, select Properties from the menu, then click the Extensions tab, which Figure 6 shows. To add a new CDP to the certificates that the CA issues, click Add. You can configure the CDP field of a CA's proper certificate by using a capolicy.inf configuration file.

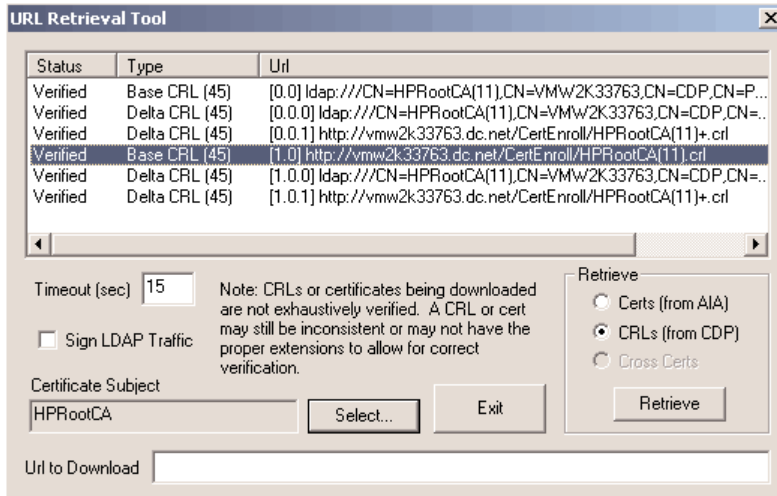
**Figure 6**  
*ResearchSubCA Properties Extensions tab*



A convenient way to test CDPs embedded in a Windows X.509 certificate extension is to use the URL Retrieval Tool, which Figure 7 shows and which comes with the Windows 2003 version of the Certutil command-line tool. To access the URL Retrieval Tool, type

```
certutil -URL <certificate_
file_name>
```

**Figure 7**  
*URL Retrieval Tool*



To retrieve complete CRLs and delta CRLs, select the *CRLs (from CDP)* option in the Retrieve section, then click Retrieve. Double-clicking one of the rows in the upper part of the URL Retrieval Tool opens the CRL viewer for the selected CRL. You can also use the URL Retrieval Tool to retrieve CA certificates mentioned in a certificate's Authority Information Access (AIA) field.

## Netscape Revocation Extensions

Netscape uses a proprietary online certificate revocation checking method. The company embeds a custom extension, *netscape-revocation-url*, in all its certificates. *Netscape-revocation-url* points to a Web page that checks the certificate revocation. To send the revocation-checking request to the Web page, Netscape uses the HTTP GET method with a URL that concatenates *netscape-revocation-url* and the serial number of the certificate that needs to be checked. The response that comes back from the Web server is a document with Content-Type *application/x-Netscape-revocation*. The document contains one digit (either 1 if the certificate isn't valid or 0 if the certificate is valid). To enable a Windows 2003 or Win2K CA to issue certificates containing this extension, you must use the Certutil tool. From the command line, type

```
certutil -setreg Policyrevocationtype +AspEnable
```

You must also restart the CA service to make this change effective.

## A Crucial PKI Service

Certificate revocation checking is a crucial PKI service, and reliable revocation checking is an important part of a trustworthy PKI service. Make sure you address revocation checking when you design your PKI environment. Both Windows 2003 and Win2K let you automate the certificate-revocation process, and Windows 2003 PKI includes some important enhancements in the revocation-checking space, so use them.

## Chapter 7:

# Windows Server 2003 PKI Key Archival and Recovery

—by *Jan De Clercq*

Key archival and recovery are public key infrastructure (PKI) services that organizations can use to recover lost, stolen, or unavailable private encryption keys. Key archival and recovery are requirements in PKI-enabled applications, such as secure mail applications, that deal with persistent data. Microsoft first introduced automatic and centralized private key archival and recovery in the Key Management Service (KMS), part of the Secure MIME (S/MIME)-based mail application in Microsoft Exchange Server 4.0 and later. (Exchange Server 2003 doesn't come with a KMS, so if you have an operational KMS in an Exchange 2000 Server environment and plan to migrate to Exchange 2003, you must migrate the KMS key archival database to the Windows Server 2003 Certification Authority—CA—key archival database.) Windows 2003 key archival and recovery build on the KMS concept: Each Windows 2003 enterprise CA includes an automated and centralized key archival and recovery service. (PKI users can also manually perform key archival and recovery, as the sidebar “Manual Key Archival and Recovery.”) Let's examine how to set up automatic Windows 2003 PKI key archival and recovery and how archival and recovery work.

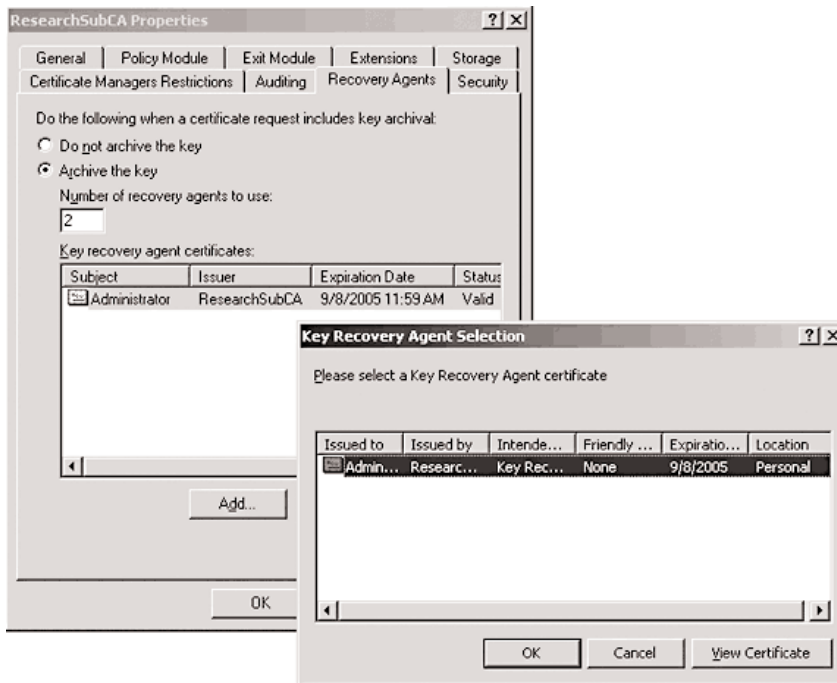
## Configuring Automatic Key Archival and Recovery

A Windows 2003 CA can automatically and securely archive PKI users' private keys in the CA database. The key archival process then occurs transparently, as part of a user's certificate enrollment process. (For more information about the certificate enrollment process, see Chapter 5.) First, however, you must make some configuration changes to the enterprise Windows CA and to your organization's certificate templates.

To configure a CA object's key archival settings, open the Microsoft Management Console (MMC) Certification Authority snap-in. Open the CA object's Properties dialog box and go to the Recovery Agents tab. Select the *Archive the key* option to enable key recovery. Like the Exchange KMS, the Windows 2003 CA supports a missile-silo system for key recovery: You can require multiple key recovery certificates and thus multiple Key Recovery Agents (KRAs) to recover one key from the archival database. (A KRA is a Windows account that owns a Key Recovery certificate and private key and therefore has key recovery privileges. Windows 2003 PKI comes with a predefined Key Recovery certificate template, so setting up a Key Recovery certificate for a particular account is relatively easy. For optimum security, you can store both the KRA's certificate and private key on a smart card.) Specify the number of KRAs necessary to recover a key in the *Number of recovery agents to use* text box. Next, select the KRA certificates that you want to use for key archival. (Note that you can add more KRA certificates than are necessary for key recovery.) To do so, click Add at the bottom of the tab. The CA logic queries the KRA container in the Active Directory (AD) Configuration naming

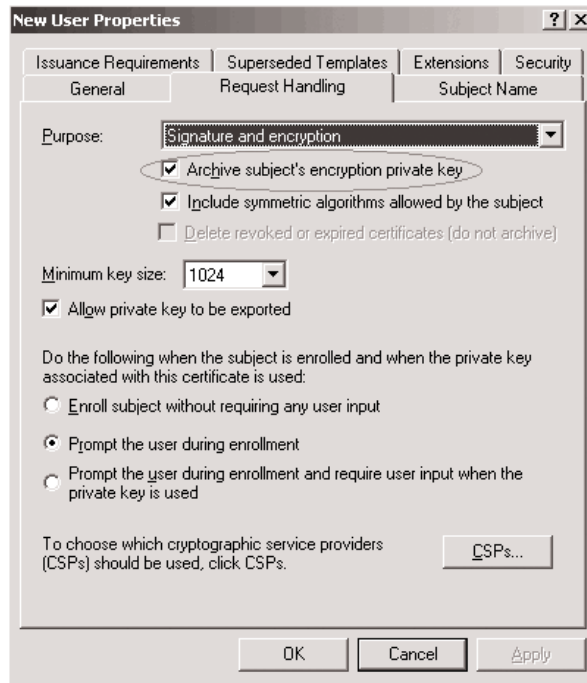
context (NC) and retrieves a list of available KRA certificates, as Figure 1 shows. Each time you add a KRA certificate, you must restart the CA service; until you do so, the certificate's Status column will read Not Loaded.

**Figure 1**  
*Key Recovery Agent Selection*



To enable key archival at the certificate-template level, open the MMC Certificate Templates snap-in. To automatically archive a PKI user's private key when the user requests a certificate based on a particular template (e.g., the New User template), open the template's Properties dialog box and go to the Request Handling tab. Select the *Archive subject's encryption private key* check box, as Figure 2 shows. Note that you can select this option only for Version 2 certificate templates.

**Figure 2**  
*New User Properties dialog box*



## Automatic Key Archival and Recovery Architecture

After you've enabled automatic key archival, for each private key archival request the CA generates a random, symmetric Triple Data Encryption Standard (3DES) key that the CA then uses to encrypt the PKI client's private key. The CA then uses the public key of the KRA that you configured for key archival to encrypt the symmetric encryption key. If you selected more than one KRA, the CA will encrypt the symmetric encryption key with each KRA's public key. Step by step, the archival process is as follows:

1. The PKI client queries AD for a CA. The client looks specifically for CA entries in the Configuration NC's Enrollment Services container. AD returns the CA's name and location.
2. The PKI client asks the CA for a copy of its CA Exchange certificate. The CA returns the certificate to the client.
3. The PKI client validates the certificate, verifying the signature, performing a revocation check, and validating the certificate format.
4. The PKI client uses the CA Exchange certificate's public key to encrypt the client's private key. The client embeds the encrypted blob by using the Certificate Management protocol with Cryptographic Message Syntax (CMS)—CMC—and forwards the CMC-formatted file to the CA.

## Manual Key Archival and Recovery

Windows public key infrastructure (PKI) users have several options for manually backing up their private encryption keys. The preferred and most commonly used format for archived private keys is a Public-Key Cryptography Standards (PKCS) #12 (.pfx) file. Users can use a password to secure access to and confidentiality of a .pfx file's content. To manually back up private keys to this type of file, a user can do one of the following:

- Use the Microsoft Management Console (MMC) Certificates snap-in to open his or her personal certificate store. Right-click the certificate of the private key the user wants to back up, then select All Tasks, Export from the context menu. This action launches the Certificate Export Wizard. The user must select both the *Yes, export the private key* and the *Enable strong protection (requires IE 5.0, NT 4.0 SP4 or above)* options. The latter option will make the wizard prompt the user for a password to protect the .pfx file's content. The user shouldn't select the *Delete the private key if export is successful* option.
- Open Microsoft Internet Explorer (IE) 6.0 and select Tools, Internet Options from the menu bar. In the Internet Options dialog box, go to the Content tab. Click Certificates to open the Certificates dialog box. Select the certificate of the private key the user wants to export, then click Export to launch the Certificate Export Wizard. Use the same options I described previously.

Users can also archive their private encryption keys from Microsoft Outlook. Outlook doesn't store the keys in a .pfx file; instead, it uses a special Outlook export (.epf) file. The .epf extension shows the historical roots of Outlook's secure mail technology: EPF stands for *Entrust profile*. One reason Outlook still uses this format is that it supports X.509 Version 1 certificates, which early Exchange Key Management Service (KMS) implementations use. As with .pfx files, users can use a password to secure .epf files.

To export private encryption keys from Outlook, a user can select Tools, Options from the Outlook menu bar, then go to the Security tab. At the bottom of the Security tab, the user should click Import\Export to open the Import\Export Digital ID dialog box. In that box, the user should select the *Export your Digital ID to a file* option, then select the Digital ID of the private key the user wants to export and fill in a filename and password. The user shouldn't select the *Delete Digital ID from system* check box.

5. The CA uses the private key associated with the CA's Exchange certificate to decrypt the client's encrypted private key, then encrypts the private key with a random 3DES symmetric key.
6. The CA determines whether the private key in the CMC-formatted file cryptographically pairs with the public key in the certificate request. The CA also validates the signature on the request by using the public key that comes with the request.
7. The CA uses the public keys of one or more KRAs (depending on the CA configuration) to encrypt the symmetric key.
8. The CA saves the encrypted blob containing the client's encrypted private key and the symmetric encryption key in the CA database.
9. The CA processes the certificate request, forwards the certificate to the user, and publishes the certificate in the directory (if that option is set in the certificate template).

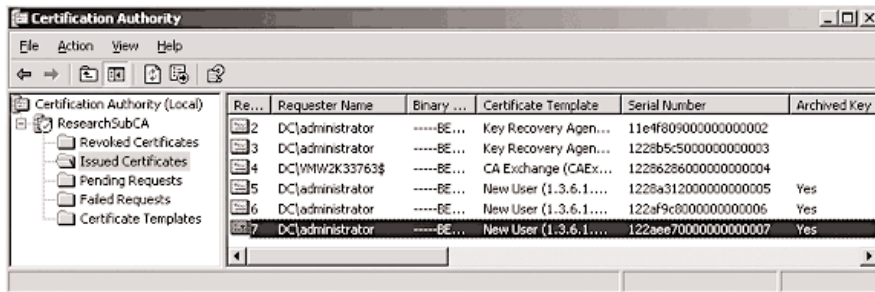
In this process, the CA Exchange certificate provides confidentiality and protects integrity when the PKI client forwards its private key to the CA for archival. Windows 2003's new certificate template defines the certificate's content. A CA Exchange certificate physically resides in the attributes of the CN=<CAName>-Xchg,DC=<domainname> AD object. The certificate's private key is in a secured part of the CA server's registry. For security reasons, the CA Exchange certificate and key pair have a short lifetime—7 days.

The Windows 2003 CA stores the client's encrypted private key in the CA database's RawArchivedKey column and stores the encrypted symmetric key in the KeyRecoveryHashes column. To view these columns and the rest of the CA database's schema from the command line, type

```
certutil -schema
```

You can also use the Certification Authority snap-in to determine whether a certificate's private key is archived in the CA database, as Figure 3 shows. To do so, you must add the Archived Key column to the display for the CA's Issued Certificates container. To do so, right-click the Issued Certificates container, select View, *Add/Remove columns* from the console's menu bar, then add the Archived Key column.

**Figure 3**  
*Certification Authority*



## Key Recovery

A PKI user or a PKI-enabled application's user typically initiates key recovery, which requires the intervention of at least one KRA (depending on the number of KRAs specified in the CA's properties). Windows 2003 PKI supports role separation—letting you separate the roles of CA administrator, certificate manager, and KRA—so key recovery might also require the intervention of a certificate manager to retrieve the recovery data from the CA database. The following examples assume no role separation and use only one KRA certificate. You can use the command line or a GUI to recover an archived private key.

A full Windows 2003 private key recovery sequence from the command line consists of the following steps:

1. The KRA identifies the user requesting a private key recovery.
2. The KRA records the user certificate's user principal name (UPN), common name (CN), account name (domain\username), Secure Hash Algorithm-1 (SHA-1) thumbprint, or serial number with

the goal of finding a unique identifier by which to identify the key. If a particular user has more than one archived key, the safest method is first to retrieve a list of all archived keys. The KRA can use the following command:

```
certutil -getkey <user CN, account name, or UPN>
```

to retrieve a list of all archived keys for the user. This command returns the serial number of each archived key; the KRA can then identify the key to recover and use the corresponding serial number as a unique identifier.

3. To export the recovery data from the CA database, the KRA opens a command prompt and types

```
certutil -getkey <unique identifier> <output file>
```

4. Next, to transform the output file to a Public-Key Cryptography Standards (PKCS) #12 file that contains the recovered private key and is secured by using the password *test*, the KRA types

```
certutil -p "test" -recoverkey <output file> <PKCS#12 file>
```

5. If the KRA recovers multiple keys for the user, the KRA can then merge the multiple PKCS #12 files into one PKCS #12 file by typing

```
certutil -p "test" -MergePFX -user "<PKCS#12_file1>,<PKCS#12_file2>"
"<NameofCombined_PKCS#12>"
```

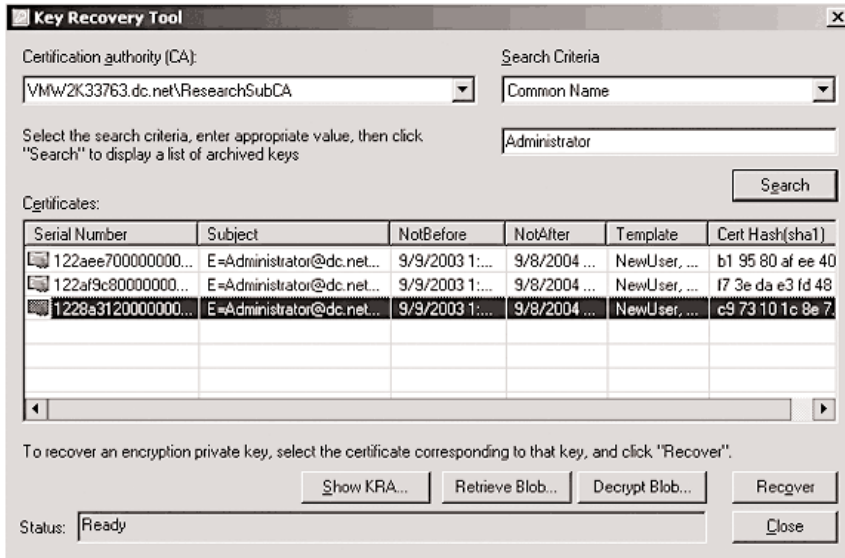
6. The KRA provides the final PKCS #12 file to the user, who can import it to his or her certificate store.

To recover keys by using a GUI, the KRA must use the *Microsoft Windows Server 2003 Resource Kit's* Key Recovery tool (krt.exe—aka Certification Authority Key Recovery), which Figure 4 shows.

To recover keys by using the Key Recovery tool, the KRA must perform the following steps:

1. In the *Certification authority (CA)* drop-down box, the KRA selects the CA from which to recover keys.
2. To search for the archived private keys and certificates for a particular user, the KRA selects a search criterion (i.e., Common Name, UPN, Serial Number, Hash, or Account Name) in the Search Criteria drop-down box, then enters the appropriate identifier (e.g., Administrator, to correspond with the Common Name criterion) and clicks Search.
3. After the search returns its results, the KRA can either click Recover to recover all archived keys at once or use the Retrieve Blob and Decrypt Blob buttons to retrieve one key-certificate pair.

**Figure 4**  
*Key Recovery Tool*



## Data Recovery vs. Key Recovery

Data recovery is a PKI-related process that decrypts encrypted data following the loss of a private key. This service is necessary when dealing with persistent data that's secured by using encryption technology. The inability to decrypt such data when an encryption key is lost would result in data loss. Data recovery can follow key recovery: After a user and authorized administrator gain access to the user's private key, the user can use the key to decrypt the encrypted symmetric keys that were used to encrypt the data.

However, data recovery can also occur independently of private key recovery. Windows 2003 or Windows 2000 Encrypting File System (EFS) is a good example of an application in which data recovery can occur independently of user private key recovery.

For cases in which data recovery must occur independently of private key recovery, a predefined set of administrators—referred to as *Data Recovery Agents*—are authorized to decrypt the data. The symmetric encryption key must be available to the Data Recovery Agents. Therefore, a PKI that uses this type of data recovery typically uses the Data Recovery Agents' public keys to encrypt a copy of the symmetric encryption key.

Do you need to support data recovery? Keep the following points in mind when making this decision:

- Data recovery is required when your organization requires independent access to users' encrypted data or doesn't permit access to users' private keys.
- Key recovery is sufficient when your organization permits access to private keys and doesn't require (or permit) independent access to users' encrypted data.

- If your organization doesn't permit access to users' encrypted data or private keys, neither key recovery nor data recovery is right for you.

## Powerful Capabilities

Windows 2003 PKI provides powerful, centralized, and automatic new key archival and recovery capabilities. The new key archival and recovery service is one of the major reasons why Windows 2003 PKI is a much more mature PKI than its predecessors and (together with other new features) will help Windows 2003 PKI better compete with PKI offerings from other vendors.

## Chapter 8:

# Using Certificates to Secure Your WLAN

—By *Randy Franklin Smith*

Without 802.1x, trying to set up and maintain a secure wireless LAN (WLAN) is a nightmare because of vulnerabilities in the Wired Equivalent Privacy (WEP) standard, especially poor key-management techniques such as manual key distribution. Although 802.1x addresses WEP's major vulnerabilities, you must configure each component to use 802.1x, including workstations, wireless Access Points (APs), and a Remote Authentication Dial-In User Service (RADIUS) server. In addition, the RADIUS server needs a credentials database that it can use to authenticate wireless clients, and you need a Certificate Authority (CA) to grant the RADIUS server a certificate for authenticating itself to wireless clients.

However, Microsoft has leveraged Active Directory (AD) and Group Policy to the point that you can completely insulate the user from the 802.1x implementation process. When your WLAN and clients are properly set up, an authorized workstation that's brought within range of your WLAN automatically authenticates and connects to the WLAN without any action by the user. Unauthorized workstations are blocked from connecting to the WLAN or snooping on its traffic. With 802.1x, there are no WEP keys to manually distribute to APs and workstations, and no lists of media access control (MAC) addresses of authorized workstations on each AP. An 802.1x WLAN first requires wireless clients to authenticate through the AP to a RADIUS server, then lets the AP and wireless client negotiate dynamic encryption keys instead of using the much weaker static keys that most WEP networks use.

I'm pretty blown away by what a good job Microsoft has done integrating 802.1x support into the Windows environment—how easy it is to set up and how well it works. I'm going to show you the simplest way to implement 802.1x and certification-based authentication on a typical network of Windows XP and Windows 2000 computers and a Win2K AD domain. Alternatively, you can use passwords for WLAN authentication. In this case, you can configure workstations so that when they come within range of your WLAN, they either use the username and password that the user specified when logging on to the workstation or prompt the user to manually enter new credentials. Password-based authentication is simpler to roll out than certificates because you don't have to create the certificates, but password authentication requires more action from the user to get on the network, and it's less secure. Password-based authentication leaves your network vulnerable to anyone who can guess an authorized user's password—and we all know how weak user passwords tend to be. Certificate-based authentication lets only users who have a computer with an authorized certificate and private key (or can steal such a computer) on the network.

Although Windows supports the most recent wireless security standard—Wi-Fi Protected Access (WPA), which uses 802.1x and addresses WEP's vulnerabilities—I don't use WPA in this chapter for several reasons. First, as I write this chapter, you can't use Group Policy to roll out the WPA update automatically to all your workstations—a major drawback if you have many workstations. Second, WPA is actually an interim standard adopted by the wireless industry until the official 802.11i is

ratified, which means that if you implement WPA now, you'll need to roll out another update relatively soon. Third, WPA requires device driver or firmware updates for your many wireless NICs and firmware updates for your APs. When you look at all the work required to implement WPA and the little extra protection WPA provides compared with how easily you can implement 802.1x and how much protection 802.1x provides, WPA just doesn't seem worth the trouble. If Microsoft provides a way to update systems and NICs automatically to 802.11i when it comes out, I think 802.11i will be a worthwhile investment.

## Adding X

To set up 802.1x on a WLAN, the first thing you need to do is make sure your network supports 802.1x. Windows Server 2003 comes with 802.1x built in, and Microsoft has added 802.1x support to XP with Service Pack 1 (SP1) and to Win2K with the Microsoft 802.1x Authentication Client available at <http://www.microsoft.com/windows2000/server/evaluation/news/bulletins/8021xclient.asp>. (You can even obtain 802.1x authentication clients for Windows NT and Windows 9x if you have a Premier or Alliance support contract with Microsoft, but you won't be able to use Group Policy to push out a centrally configured wireless networking policy to those computers.)

Next, make sure that any APs you currently have or plan to purchase support 802.1x. Typically, 802.1x-compliant APs have an 802.1x configuration page that you can find when you log on to the AP through your Web browser.

Finally, you must set up one Windows 2003 server and install Internet Authentication Service (IAS) on it. IAS provides the RADIUS server necessary on an 802.1x WLAN. When a wireless client tries to connect to an AP, the AP contacts the RADIUS server to try to authenticate and authorize the client. The RADIUS server checks the client's credentials against AD and lets the AP know whether to let the wireless client connect. You need to use Windows 2003's IAS instead of Win2K Server's IAS because only Windows 2003's IAS supports 802.1x authentication services. Make sure that the Windows 2003 computer that will serve as the IAS server is a member of the domain but not a domain controller (DC). Then open the Control Panel Add/Remove Programs applet, select Add/Remove Components, and install Internet Authentication Service.

## Certificate Services

The IAS server needs a certificate to authenticate itself to the wireless clients, so you need a CA server running either Windows 2003 or Win2K Certificate Services. The CA server will also provide certificates for all your wireless clients for authentication to the IAS server. If your AD domain is still running on Win2K DCs and you don't already have an enterprise CA, I suggest that you install Certificate Services on an existing Win2K server as an Enterprise Root CA instead of installing Certificate Services on your Windows 2003 IAS server.

When you install Certificate Services, you're basically setting up a public key infrastructure (PKI). For a large enterprise PKI, you should set up one root CA that you then use to certify other subordinate CAs. You keep the root CA off the network to ensure that it's never compromised, and you issue certificates from the subordinate CAs. If a subordinate CA is ever compromised, you can revoke its certificate from the root CA and publish a new certificate revocation list (CRL) to your network. A multilevel CA hierarchy saves you from rebuilding your entire PKI if a CA is compromised. However, it can be overkill for smaller networks. For the sake of simplicity, we'll just set up one Enterprise Root CA on an existing Win2K server and issue certificates directly from it.

Open the Add/Remove Programs applet, then select Add/Remove Components. Before installing Certificate Services, you must install Microsoft IIS because Certificate Services uses IIS for administration and certificate requests. When you install IIS from Control Panel, be sure to enable Active Server Pages (ASP) so that the Certificate Services pages will work. After installing IIS, you can install Certificate Services. When prompted for what type of CA to install, select Enterprise Root CA. “Enterprise” means that CS will automatically integrate with AD, letting you enroll users and computers automatically.

Next, you’re prompted for identifying information for the CA. I use the name STO CA in this chapter. After you set up the Enterprise Root CA, all the computers in your domain will automatically trust certificates issued by that CA. Why? When the computers see that a new enterprise CA is published in AD, the computers automatically add that CA’s certificate to their Trusted Root Certification Authorities store, which you can view with the Microsoft Management Console (MMC) Certificates snap-in.

## Obtaining Certificates

Now that you have a CA, you need to obtain a certificate for your IAS server. Open the Certificates snap-in on the IAS server. When prompted to choose a user, service, or computer account, select computer; when prompted to choose between a remote computer or the local computer, choose the local computer. Then in the main Certificates snap-in window, right-click the Personal\Certificates folder and select All Tasks, Request New Certificate. Click Next on the first page of the Certificate Request Wizard. On the next page, select Computer as the certificate template type, and click Next. Enter WLAN Authentication as the friendly name, click Next, then click Finish. A dialog box will tell you that the certificate request was successful; then, you should see the new certificate in the Personal\Certificates folder.

Now you need to obtain certificates for all your wireless client computers. The easiest way to do so is to create a certificate request in a Group Policy Object (GPO) that will be applied to all your computers that need WLAN access. Either create a new GPO that’s linked to an appropriate organizational unit (OU) or edit the Default Domain Policy GPO that’s linked to the root of the domain (in most environments, there’s no harm in giving each computer in the domain a Computer certificate). In Group Policy Object Editor, navigate to the GPO’s Computer Configuration\Windows Settings\Security Settings\Public Key Policies\Automatic Certificate Request Settings node. Right-click in the details pane and select New, Automatic Certificate Request. Click Next on the first page of the wizard. On the next page, select the Computer template, click Next, then click Finish. You should now see a request for a Computer certificate in the Automatic Certificate Request Settings folder. The next time each computer applies this GPO, the computer will request a Computer certificate from your CA and store it in the computer’s personal certificate store.

## Wireless Client Network Settings

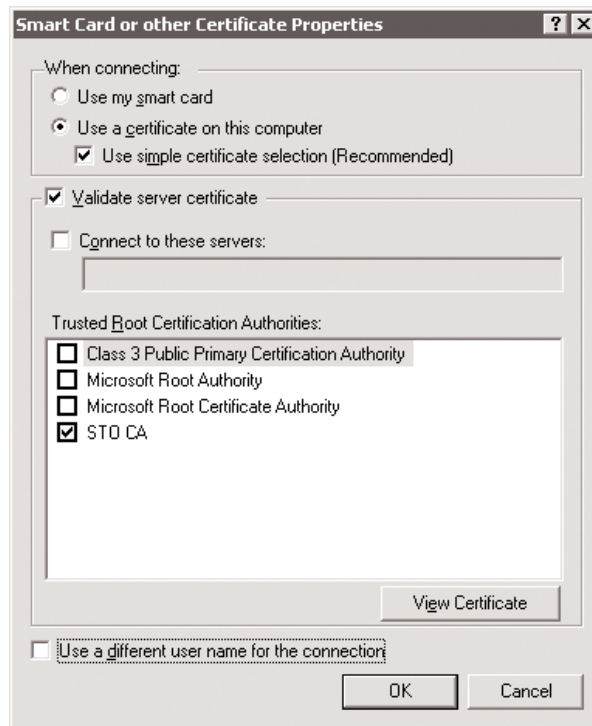
The next step involves configuring the wireless network settings on all your wireless client computers. Again, you’ll use Group Policy to automate the process. Using the same GPO as before, navigate to Computer Configuration\Windows Settings\Security Settings\Wireless Network (IEEE 802.1x) Policies. Right-click in the details pane and select Create Wireless Network Policy. Click Next on the first page of the Wireless Network Policy Wizard, and enter 802.1x Computer Certificate WLAN Policy as the name of the policy. Click Next, select *Edit policies*, and click Finish.

In the 802.1x Computer Certificate WLAN Policy Properties dialog box, go to the Preferred Networks tab, where you'll enter the Service Set Identifier (SSID) of your WLAN and configure its authentication settings. Click Add, and enter AcmeSecureWLAN as the network name (SSID). Now navigate to the IEEE 802.1x tab. Select *Smart Card or other certificate* as the Extensible Authentication Protocol (EAP) type and *Computer only* as the computer authentication type. Next, you need to configure the certificate settings, so click Settings.

In the *Smart Card or other Certificate Properties* dialog box, select *Use a certificate on this computer* and *Use simple certificate selection (Recommended)*, which Figure 1 shows. These two settings cause wireless clients to select an appropriate certificate (the one we just deployed) from their personal store and use the certificate to authenticate to the IAS server. To make sure the clients authenticate to the real WLAN and not some impostor WLAN set up by an attacker, select *Validate server certificate* in the same dialog box—otherwise, your wireless clients will trust any WLAN that claims to be AcmeSecureWLAN. Then, select the box next to the CA that issued your IAS server a certificate—in this case, STO CA. Click OK three times to close all the dialog boxes. Now, the next time each computer applies this GPO, the computer will configure its wireless network settings as we just defined.

**Figure 1**

*Smart Card or other Certificate Properties dialog box*



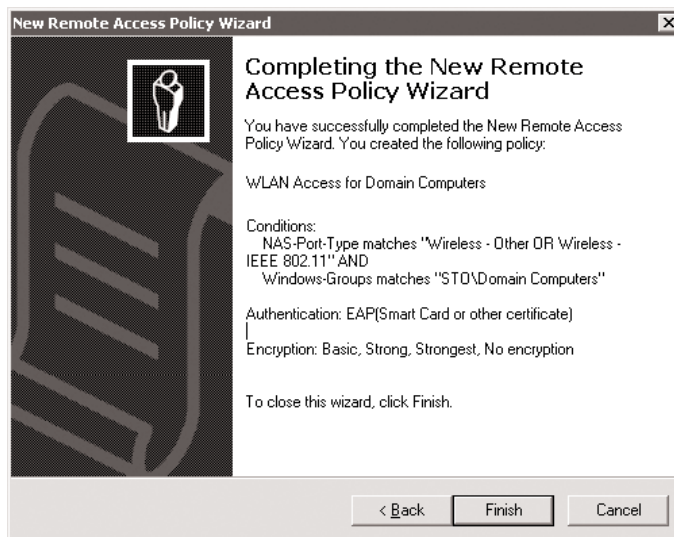
## Configure IAS and the APs

Now that you've configured your wireless clients, you're ready to configure your IAS server and APs. On the IAS server, you need to define a remote access policy that links WLAN authentication requests to a requirement for certificate-based 802.1x authentication. You also need to create RADIUS client records for each of the APs on your WLAN. To create the remote access policy, open the MMC Internet Authentication Service snap-in and select the Remote Access Policies folder. Right-click in the details pane and select **New Remote Access Policy**. Click **Next** on the first page of the **New Remote Access Policy Wizard**. On the next page, select *Use the wizard to set up a typical policy for a common scenario*, enter *WLAN Access for Domain Computers* as the name of the remote access policy, and click **Next**. On the **Access Method** page, select **Wireless** to set up the remote access policy with the necessary RADIUS criteria to recognize WLAN authentication requests and apply this policy to those requests. Click **Next**.

On the *User or Group Access* page, you define which computers can connect to the WLAN. Select **Group** and add the **STO\Domain Computers** group. **STO\Domain Computers** includes all computers in the domain. You could use a less inclusive group, but don't forget that you're also going to add a certificate requirement, so when you're finished, only computers that are a member of **STO\Domain Computers** and have a certificate trusted by the IAS server will be able to connect. Click **Next**. At the **Authentication Methods** page, you can choose from **Protected EAP (PEAP)** or *Smart Card or other certificate*; select the latter and click **Configure**. In the *Smart Card or other Certificate Properties* dialog box, you can choose which certificate the IAS server will use to authenticate itself to the client. Select the certificate that your CA issued to the IAS server, and click **OK**. Back in the wizard, click **Next**, which brings you to a page that summarizes the remote access policy's details, as Figure 2 shows. Click **Finish**.

**Figure 2**

*New Remote Access Policy Wizard*



Next you need to configure the RADIUS client records for your APs so that IAS will recognize them. Still in the Internet Authentication Service snap-in, right-click RADIUS Clients and select New, RADIUS Client. In the New RADIUS Client wizard, enter a friendly name for one of your APs and its IP or DNS address. For this example, I'll call the AP First Floor East and give it the IP address 192.168.100.3. Click Next. On the Additional Information page, select RADIUS Standard as the Client-Vendor. The other information you need to enter on this page is a secret to be shared by the IAS server and the AP. Using this secret, the two systems will be able to authenticate to each other and encrypt data sent between them. Enter the same long, complex string of characters in both the *Shared secret* and *Confirm shared secret* fields, and remember or write down the secret for when you configure the AP. Select the *Request must contain the Message Authenticator attribute*, which causes the IAS server to require the AP to use the shared secret, and click Finish. Repeat this process (with different friendly names and addresses) for the other APs.

The final setup step is to configure your APs to communicate with IAS via RADIUS and to require 802.1x authentication from wireless clients. Most AP vendors provide a tiny Web server on their APs so that you can configure them from a Web browser on your workstation. Each AP's configuration pages will look a little different, but they all require the same basic settings for enabling 802.1x. On your AP, find the page that lets you enable 802.1x authentication. Next, enter the IP address of your IAS server and the shared secret. If the AP asks for a port number, enter 1812. Your AP might also let you enter a second RADIUS server, if available, for fault-tolerance purposes. Finally, configure the AP with the appropriate SSID.

## Test Case

Now that everything is set up, it's time to test. First, test WLAN connectivity using a legitimate client—a computer that has the appropriate wireless network policy and a computer certificate from the CA. The computer should connect to the WLAN automatically, and a balloon should appear in the lower right corner of the desktop to indicate a wireless connection. Next, try connecting to the WLAN with an unauthorized client. Make sure the WLAN blocks it. You could also obtain a WLAN sniffer and confirm that your WLAN traffic is indeed encrypted.

If you run into any problems, you can diagnose them from the wireless client, the AP, or the IAS server. On the client, check the System and Security logs for certificate-related error messages. Most APs have a logging feature that you can use for tracking down authentication problems between the client and IAS or RADIUS problems between the AP and IAS. On the IAS server, you can look for errors in the System log as well as in the IAS log at C:\windows\system32\logfiles. The IAS log isn't very readable, but Iaspase does a great job of producing readable reports from the IAS log.

## Chapter 9:

# FAQs

## Obtaining a Server Certificate from Your Own CA

—by Randy Franklin Smith

**Q**I need to set up a secure extranet Web server so that we can exchange information with clients and contractors. When I try to configure Microsoft Internet Information Services (IIS) to use HTTPS for the site, IIS requires me to install a server certificate. I'm embarrassed to admit that management doesn't want to spend a couple hundred dollars to buy a Web Secure Sockets Layer (SSL) certificate from a third-party Certification Authority (CA). Can I set up SSL on my server without installing a server certificate?

**A** SSL requires a server certificate. However, you can set up your own CA, then obtain a certificate from it.

To set up a CA, simply install Certificate Services on an available server. Then, disconnect your Web server from the Internet and reconnect it to your internal LAN. (Although there is a way to request a certificate offline, I find this method easier unless I'm dealing with a server that hosts sites and that can't be taken down.) Open Microsoft Internet Explorer (IE), and enter the address of your CA followed by /certsrv (e.g., <http://ca/certsrv>) in the Address bar. IE will display the Microsoft Certificate Services Web page.

To request a certificate from your new CA, click the *Request a certificate* link. On the *Request a Certificate* page, click the *advanced certificate request* hotlink. Finally, click the *Create and submit a request to this CA* link. Be sure to enter the Web server's DNS name or IP address (depending on how you access the server from the Internet) in the Subject field. For *Type of Certificate Needed*, select Server Authentication or Computer, depending on which one appears. You can leave the other settings at their defaults.

After requesting the certificate from your Web server, log on to your CA and approve the request, which you'll find in the Pending Requests folder in the Microsoft Management Console (MMC) Certification Authority snap-in. Back at your Web server, browse to your CA again and click the *View the status of a pending certificate request* link. On the resulting page, you'll see the Web server's new certificate. When you install the certificate, install it in the computer's certificate store (not in your own certificate store) under Personal\Certificates.

Browse again to your CA's /certsrv page. This time, click *Download a CA certificate, certificate chain, or CRL*, click the Download CA Certificate link, and save the certificate to your Web server in a folder under Inetpub so that it will be accessible to Web-site users. Next, create a link in an appropriate place on your Web site so that users can install your CA's self-signed certificate as a trusted CA. Finally, configure IIS to use your Web server's new certificate (not the CA's self-signed certificate).

Now, when users browse to <https://yourserver.com>, they'll see a warning that your Web server is presenting a certificate signed by an untrusted CA. Instruct the users to let the Web page load proceed. After users log in to your site, have them click the link to your CA's certificate and let IE install the certificate in the Trusted Root Certification Authority store. Thereafter, users should no longer see the warning.

A word of caution: Keep the system your CA resides on secure. If you don't plan to issue new certificates in the near future, it's a good idea to disable the Certificate Services service.

## Using Windows Server 2003's Certificate Templates

—by *Randy Franklin Smith*

**Q**I'm playing around with Windows Server 2003's Certificate Services in preparation for upgrading our Windows 2000 Certification Authorities (CAs). I've noticed many new certificate templates in the Windows 2003 Microsoft Management Console (MMC) Certificate Templates snap-in, but I can't enable them. When I open the MMC Certification Authority snap-in, right-click the Certificate Templates folder, then click New, *Certificate Template to Issue*, I see only a subset of the templates that are available in the Certificate Templates snap-in. Where are the rest of the templates, and why can't I issue them from this CA?

**A**Evidently you're testing Windows 2003, Standard Edition or Windows 2003, Web Edition. Microsoft significantly enhanced certificate templates in Windows 2003 but in effect charges a premium to use that functionality by enabling it only for Windows 2003, Enterprise Edition and Windows 2003 Datacenter Edition.

Windows 2003 offers several new certificate templates that give you more versatility and finer control over the properties that constitute a certificate. Moreover, you can duplicate the default certificate templates and customize them to your needs. For example, you can control the intended purposes (e.g., Server Authentication, Client Authentication, encryption, digital signature) for certificates issued by a given template. You can also control the issuance policy for each template to allow some templates to be issued automatically without CA administrator approval whereas other templates require administrator authorization.

Windows 2003 also includes a new feature called Autoenrollment. Traditionally, when you wanted to deploy a certain type of certificate to a set of users or computers, you had to configure one or more Group Policy Objects (GPOs) in Active Directory (AD) with an Automatic Certificate Request setting (under Computer Configuration\Windows Settings\Security Settings\Public Key Policies in any GPO) that directed the users or computers to request a certificate according to the associated template. With Autoenrollment, you can simply add the desired template to your CA's Certificate Templates folder. After you do so, the ACL will automatically request the new certificate for all computers and users who have Enroll permission on the templates—you don't need to configure Group Policy.

To control which computers or users will request the certificate template, simply open the Certificate Templates snap-in, then open the desired template's Properties page. Click the Security tab and grant Enroll permission to the user accounts or computers that you want to enroll. If you check

the Certificate Templates snap-in's Minimum Supported CAs column, you'll notice that certificates that support customization and Autoenrollment can be issued only by Windows 2003 Enterprise or Windows 2003 Datacenter CAs. You can issue all other certificates from Win2K and later servers. You'll also notice that Autoenrollment works only for new clients, such as Windows 2003 and Windows XP clients.

## Enabling SSL on Your Site

—by Brett Hill

**Q**We're trying to enable Secure Sockets Layer (SSL) on our site. We've installed a certificate but we can't create a secure (i.e., <https://>) connection. The site works fine with http, but when we use HTTP Secure (HTTPS), the Web browser waits for a long time, then times out and says it can't reach the server.

**A**Troubleshooting SSL connection problems can be tedious, but here are a few common causes that you should look for:

- Port 443 isn't configured for the SSL connection—Open the Web site properties and ensure that the number 443 is listed in the SSL Port box. A bug in IIS can cause this number to not appear in the SSL Port box.
- The IP address listed in IIS isn't assigned to a NIC—Because you can type in any IP address in the IIS UI, the IP address listing can get out of sync with the IP addresses on the NIC. From a command prompt, run

```
netstat -an
```

and make sure an IP address is listening on port 443 (the output should list 0.0.0.0:443 or the IP address you defined for that site). For more information, see the Microsoft article "HOWTO: Determine If SSL Connectivity Is Not Working on the Web Server or on an Intermediate Device".

- The default Web site is using the IP address and port 443 that you are trying to use on your Web site—Port 443 might still be bound to the default site when you try to use it on another Web site, even if you remove the certificate from the default Web site. For more information, see the Microsoft articles "Page Cannot Be Displayed When You Connect Through SSL" and the Microsoft article "IIS Binds To All Available IP Addresses When It Starts".
- Another product or service is using port 443, making that port unavailable to IIS—You can use freeware products such as SmartLine's Active Ports or Sysinternals' TCPView to show which processes are using which ports on your server.
- IP Security (IPSec) is blocking port 443—If you use IPSec rules locally on the server, IPSec might block port 443. To check this setting, open the Local Security Settings console in Administrative Tools and review the IPSec rules to determine whether a policy in-force could be the problem.
- A firewall or router is blocking port 443—Try creating an SSL session on the IIS server. If you can successfully establish this session, you probably have a firewall or router that's blocking port 443 on the network or a proxy server that isn't forwarding correctly.

- The Web site uses host headers—Remember that you can't use host headers with an SSL site. If you can use the IP address, but not the host name, to create an SSL connection to the Web site, you're probably using host headers on the Web site. For more information, see the Microsoft article "HTTP 1.1 Host Headers Are Not Supported When You Use SSL".
- The certificate isn't the proper kind of certificate—Double-click the certificate file you installed on the Web site, and make sure it says *Ensures the identity of a remote computer* or *Server Authentication* under *This certificate is intended to*. For more information, see the Microsoft article "Error Message: The Page Cannot Be Displayed ... Cannot Find Server or DNS Error".

## Using the SSL Protocol to Secure HTTP Basic Authentication Traffic

—by *Jan De Clercq*

### **Q** Should I combine Basic authentication with a Secure Sockets Layer (SSL) tunnel to protect user credentials that are sent across an HTTP connection?

**A** When you use Basic authentication, the credential information that travels between the Web browser and the server isn't secured; it's just base64 encoded. Intruders can easily decode base64; for a demonstration, you can use an online base64-decoder tool. You can find a good example of such a tool at <http://www.robertgraham.com/tools/base64coder.html>. Go to the URL, enter the Basic authentication string

```
ZG9tYWluXHVzZXJlOnBhc3N3b3Jk
```

into the decoder, and click Decode.

Because decoding base64 is so easy, I recommend that you use SSL to secure HTTP traffic. To use SSL, you'll need to set up certificates on the Web server and, optionally, on the client. SSL uses these certificates to provide the following security services:

- Server authentication—SSL uses X.509 server certificates to authenticate the Web server.
- Data confidentiality and integrity services—SSL always provides channel-encryption services.
- Optional client authentication—SSL uses X.509 client certificates to authenticate clients.

## Addressing ActiveX Controls

—by *John Savill*

### **Q** When users request certificates from a Windows Server 2003-based Certificate Authority (CA), why does the CA prompt them to download an ActiveX control?

**A** Windows 2003 includes a new version of `xenroll.dll` (an ActiveX control that can create certificates) that prompts users of previous Windows versions to download an ActiveX control when requesting a certificate. To resolve this problem, go to the Windows 2003 certificate server and perform the following steps:

1. Log on as an Administrator.
2. Open `\\%systemroot%\system32\certsrv\certdat.inc` in a text editor.
3. Locate the `sXEnrollVersion="5,131,3686,0"` entry, then modify the entry to  

```
sXEnrollVersion="5,131,3659,0"
```
4. Save the changes, then close the text editor.

## Enabling SSL on IIS

—by *John Savill*

### **Q**How can I obtain a certificate so that I can enable Secure Sockets Layer (SSL) on my Microsoft IIS server?

**A** Before you can use SSL for an IIS server, you must obtain a certificate. To request a certificate from your Certificate Authority (CA), perform the following steps:

1. Start IIS Manager—click Start, Programs, Administrative Tools, Internet Information Services (IIS) Manager.
2. Expand the Web sites, right-click the Web site for which you want to request a certificate (e.g., Default Web Site), and click Properties.
3. Click the Directory Security tab.
4. In the “Security communications” section, click Server Certificate.
5. In the Web Server Certificate Wizard, click Next.
6. Select the option “Create a new certificate” and click Next.
7. Fill in the necessary details to request a certificate.

## IIS Client Service Mapping

—by *Jan De Clercq*

### **Q**Microsoft IIS supports an access control feature known as client certificate mapping. What’s client certificate mapping, and how can I configure it?

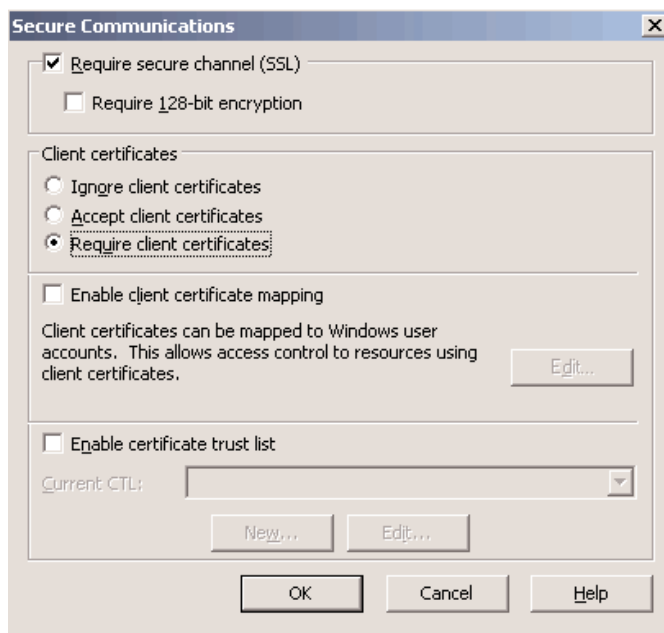
**A** When you authenticate a user who logs on to your Web server with a Secure Sockets Layer (SSL) or X.509 client certificate, you can map the information in that certificate to a Windows security identity (i.e., a Windows user account) and apply access control settings defined for that identity. Microsoft calls this feature client certificate mapping.

Client certificate mapping is available only if you’ve enabled SSL to secure access to your Web site. You can use the Microsoft Management Console (MMC) Internet Services Manager (ISM) snap-in to configure SSL. To access the SSL configuration options, right-click your Web site in the snap-in,

select Properties, select the Directory Security tab, then select the Edit button that appears in the Secure communications section at the bottom of the tab. The Edit button will appear only if you've successfully installed an SSL server certificate on your Web server.

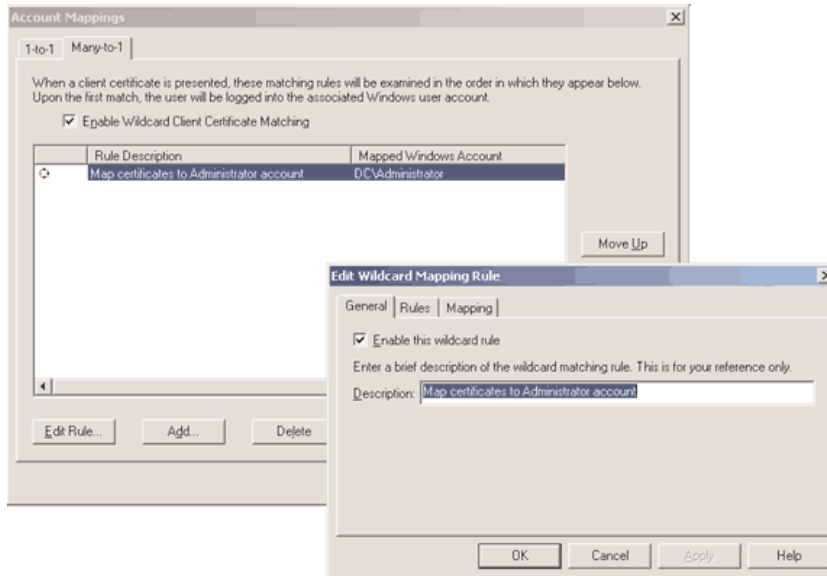
You define client certificate mapping either in the IIS metabase or in Active Directory (AD). You can enable IIS metabase-based client certificate mapping from the ISM Secure Communications dialog box, which Figure 1 shows, by selecting the Enable client certificate mapping check box. The Edit button next to this option becomes available only after you've selected this check box.

**Figure 1**  
*ISM Secure Communications dialog box*



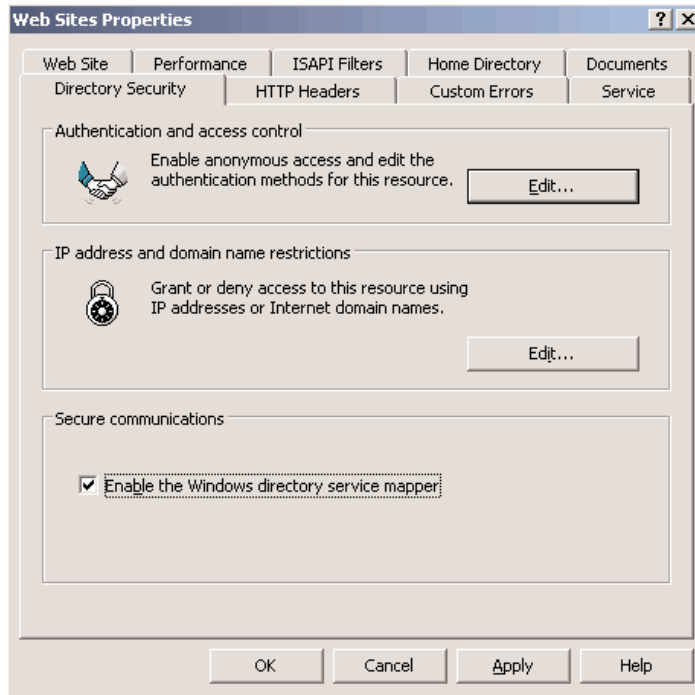
You can set up client certificate mappings defined in the IIS metabase in one of two modes: 1-to-1 mapping and many-to-1 mapping. When you use 1-to-1 client certificate mapping, IIS looks at the complete contents of the client certificate to map it to a Windows security identity. With many-to-1 client certificate mapping, IIS looks at particular attributes of the client certificate, as defined by rules that you create, to map the certificate to a Windows security identity. Figure 2 shows the dialog box for creating these many-to-1 rules.

**Figure 2**  
*Creating many-to-one rules*



AD-based client certificate mapping uses a service known as the Windows directory service mapper, which you can define from the Active Directory Users and Computers snap-in: Right-click an account object and select Name mappings (this option will be available only if the snap-in is in Advanced Features viewing mode). AD-based client certificate mapping allows for only 1-to-1 mapping. To enable AD-based mapping, open the ISM snap-in, right-click your Web site, select Properties, select the Directory Security tab, then select the Enable the Windows directory service mapper in the Secure communications section at the bottom of the tab, as Figure 3 shows. AD-based client certificate mapping is a good option if you have multiple Web servers that all need to have client certificate mappings defined. Instead of defining the mappings on each Web server, you can define them once in the central AD repository.

**Figure 3**  
*Directory Security tab*



## Controlling Which CAs Windows Can Trust

—by *Randy Franklin Smith*

**Windows arbitrarily trusts many Certification Authorities (CAs) by default. But we don't want Microsoft deciding for us which CAs' security standards and procedures are trustworthy. Is there a way to take control of which CAs Windows trusts? I know I can configure computers individually through the Microsoft Management Console (MMC) Certificates snap-in, but what if I need to change hundreds of workstations?**

**A** If your computers are running Windows 2000 or later and belong to an Active Directory (AD) domain, you can use Group Policy to mandate which CAs Windows can trust. First, open an appropriate Group Policy Object (GPO) that applies to your desired set of computers and users, then navigate to Computer Configuration\Windows Settings\Security Settings\Public Key Policies. Right-click the Trusted Root Certification Authorities subfolder and select Properties. Under *Client computers can trust the following certificate stores*, note that by default the radio button next to *Third-Party Root Certification Authorities and Enterprise Root Certification Authorities* is selected. Change the selection by clicking the radio button next to *Enterprise Root Certification Authorities*.

When next applied, Group Policy will exclude third-party root certification authorities from its trusted CAs. Although you might still see these CAs in the Trusted Root Certification Authorities store on local computers, you can prove that the computer doesn't trust a third-party CA by browsing to a site that has a certificate issued by that CA. You should receive a message to the effect that either Windows can't verify the certificate as being from a trusted CA or it can't check the certificate's revocation status. (It's a good idea to remove certificates that are unfamiliar or that you don't trust.)

## Mitigating a Problem with Computer-Only Authentication to a WLAN

—by *Randy Franklin Smith*

**Q I realize that using 802.1x and Wi-Fi Protected Access (WPA) is the best approach for securing a wireless LAN (WLAN). But are there alternative approaches that mitigate the risk of an attacker capturing information sent between legitimate wireless clients or connecting to our network and attacking computers on it?**

**A** There are alternatives to 802.1x and WPA. One approach is to set up an internal VPN server and connect it to your wireless Access Points (APs). Then, configure your wireless clients with a VPN connection and train your users to initiate the VPN connection after they obtain a wireless connection to your LAN.

A quicker, more transparent alternative uses the Microsoft Management Console (MMC) IP Security Policies snap-in and a trick with your DHCP addresses. First, reserve a range of IP addresses for your wireless clients and configure your DHCP servers to skip that range. Next, enable your AP's DHCP server feature and configure the AP to dole out addresses from that range. Then, open a Group Policy Object (GPO) that's applied to all your computers, such as the Default Domain Policy GPO. Navigate to Computer Configuration\Windows Settings\Security Settings\IP Security Policies and open the properties of the Secure Server (Require Security) GPO. This default GPO intercepts all traffic and requires it to be encrypted and authenticated via Kerberos using the computer's domain account. Change the GPO's IP filter so that instead of catching all IP traffic, the filter catches only packets to and from IP addresses that are in the range assigned to your wireless clients. Close the GPO, then right-click it and select Assign.

Now, all your computers will reject connections from wireless clients unless the clients can authenticate via Kerberos using the computer's domain account. Computers on the wired LAN will be able to use IP Security (IPSec) to communicate as before. Computers that aren't part of your domain but that try to access one of your servers will fall flat on their face. As an added precaution, consider configuring your switch to block wireless clients that try to use an IP address that's outside the range of addresses you've assigned to your wireless clients.

## Setting Up SSL Certificates for an NLB Cluster

—by *Paul Robichaux*

**Q I want to set up a Network Load Balancing (NLB) cluster for my front-end servers. What kind of Secure Sockets Layer (SSL) certificates will I need to use on those servers?**

**A** Any properly formed SSL certificate will work. If you want users to be able to use one machine name to connect to any server in the cluster, you'll need to create a certificate that has a common name (CN) that matches the NLB cluster name. Note that some third-party certificate issuers, such as VeriSign, have license agreements that prohibit you from installing the same certificate on multiple servers, so be sure to review your Certificate Authority's (CA's) license to determine how many certificates you'll need.