

# Windows Server® 2008 PKI and Certificate Security

*Brian Komar*

**PREVIEW CONTENT** This excerpt contains uncorrected manuscript from an upcoming Microsoft Press title, for early preview, and is subject to change prior to release. This excerpt is from *Windows Server® 2008 PKI and Certificate Security* from Microsoft Press (ISBN 978-0-7356-2516-7, copyright 2008 Brian Komar, all rights reserved), and is provided without any express, statutory, or implied warranties

To learn more about this book, visit Microsoft Learning at  
<http://www.microsoft.com/MSPress/books/9549.aspx>

**Microsoft®**  
Press

978-0-7356-2516-7

© 2008 Brian Komar. All rights reserved.

# Table of Contents

## 2 Primer to Public Key Infrastructure

- Certificates
- Certification Authorities
- Revocation
- Revocation Methods
- Certificate Revocation Lists
- OCSP
- Case Study: Inspecting an X.509 Certificate
- Additional Information

## Chapter 2

# Primer to PKI

## Certificates

Certificates provide the foundation of a public key infrastructure (PKI). These are electronic credentials, issued by a certificate authority (CA), that are associated with a public and private key pair.

A certificate is a digitally signed collection of information roughly two to four KB in size. A certificate typically includes the following information:

- Information about the user, computer, or network device that holds the private key corresponding to the issued certificate. The user, computer, or network device is referred to as the subject of the certificate.
- Information about the issuing CA.
- The public key of the certificate's associated public and private key pair.
- The names of the encryption and/or digital signing algorithms supported by the certificate.
- A list of X.509 version 3 extensions included in the issued certificate.
- Information for determining the revocation status and validity of the certificate.

The CA must ensure the identity of the requestor before issuing a certificate. Identity validation can be based on the user's security credentials or might include a face-to-face or in-person interview to validate requestor identity. Once identity is confirmed, the CA issues the certificate and digitally signs the certificate with its private key to prevent content modification.

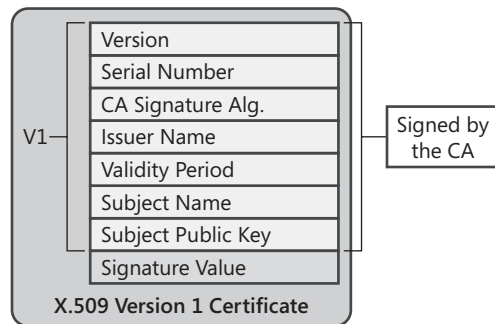
It is nearly impossible for another user, computer, network device, or service to impersonate the subject of a certificate because impersonation requires access to the certificate holder's private key. Impersonation is not possible if an attacker has access to the certificate only.

Three versions of digital certificates can be used in a PKI:

- X.509 version 1 certificates
- X.509 version 2 certificates
- X.509 version 3 certificates

### X.509 Version 1

The X.509 version 1 certificate definition was defined in 1988. Its advanced age means you rarely see version 1 certificates in networking. The exceptions are some of the older root certificates and older Exchange Key Management Service (KMS) deployments. The X.509 version 1 format defines the certificate fields, as shown in Figure 2-1.



**Figure 2-1** The X.509 version 1 certificate fields

An X.509 version 1 certificate contains the following fields:

**Version.**

- Contains a value indicating that the certificate is an X.509 version 1 certificate.

**Serial Number.**

- Provides a numeric identifier that is unique for each CA-issued certificate.

**CA Signature Algorithm.**

- The name of the algorithm the CA uses to sign the contents of a digital certificate.  
Figure 2-1 shows the fields included when creating the digital signature.

**Issuer Name.**

- The distinguished name of the certificate's issuing CA. Typically, the distinguished name is represented in an X.500 or distinguished name format specified in the X.509 specification and Request for Comment (RFC) 3280, "Internet X 509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile."

**Validity Period.**

- The range of time for which the certificate is considered valid. In some offerings, the validity period is split into two fields: Valid From and Valid To.

**Subject Name.**

- The name of the computer, user, network device, or service represented by the certificate. Typically, the subject name is represented in an X.500 or distinguished name format specified in the X.509 specification, but it can include other name formats, such as an RFC 822, "Standard for the Format of ARPA Internet Text Messages," e-mail name format.

**Subject Public Key Info.**

- The public key of the certificate holder. The public key is provided to the CA in a certificate request and is included in the issued certificate. This field also contains the public key algorithm identifier, which indicates which public key algorithm is used to generate the key pair associated with the certificate.

**Signature Value.**

- Contains the signature value that results from the CA signature algorithm used to sign the digital certificate.

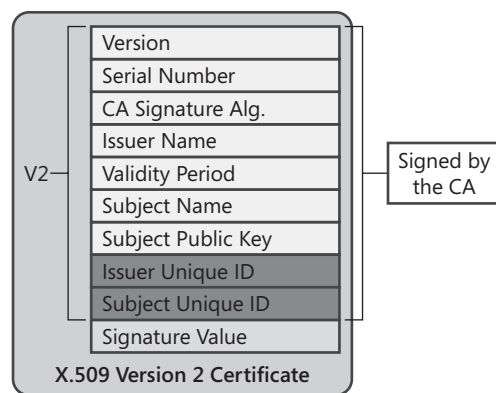
In a version 1 certificate, the Issuer Name and Subject Name fields allow certificates to be organized into a chain of certificates that starts at the certificate issued to a user, computer, network device, or service and terminates with a root CA certificate.

Certificate chaining is fully discussed in Chapter 10, "Certificate Validation."

## X.509 Version 2

While the X.509 version 1 certificate format provides basic information about the certificate holder, the format offers little information about the certificate issuer. By including only the issuer, issuer name, CA signature algorithm, and signature value, the version 1 format does not provide any provisions for CA renewal.

When a CA's certificate is renewed, two certificates possess the same Issuer Name field value. Likewise, it is possible for another organization to create a CA with the same issuer name. To address this, the X.509 version 2 certificate format was introduced in 1993. The version 2 format introduced two new fields to the certificate, as shown in Figure 2-2.



**Figure 2-2** The X.509 version 2 certificate fields

The X.509 version 2 certificate format introduced the following fields:

### Issuer Unique ID.

- An optional field that contains a unique identifier, typically a hexadecimal string, for the issuing CA as defined by the issuing CA. When a CA renews its certificate, a new Issuer Unique ID is generated for that certificate version.

### Subject Unique ID.

- An optional field that contains a unique identifier, typically a hexadecimal string, for the certificate's subject as defined by the issuing CA. If the subject is also a CA, this unique identifier is placed in the Issuer Unique ID.

In addition to introducing the Issuer Unique ID and Subject Unique ID fields, the X.509 version 2 certificate's Version field changed to a value of 2 to indicate the version number.

The Issuer Unique ID and Subject Unique ID fields improved the certificate chaining process. The process now finds the CA certificate by matching the issuer name in the issued certificate to the subject name in the CA certificate and performs a second check by matching the Issuer Unique ID in the issued certificate with the Subject Unique ID of the CA certificate.

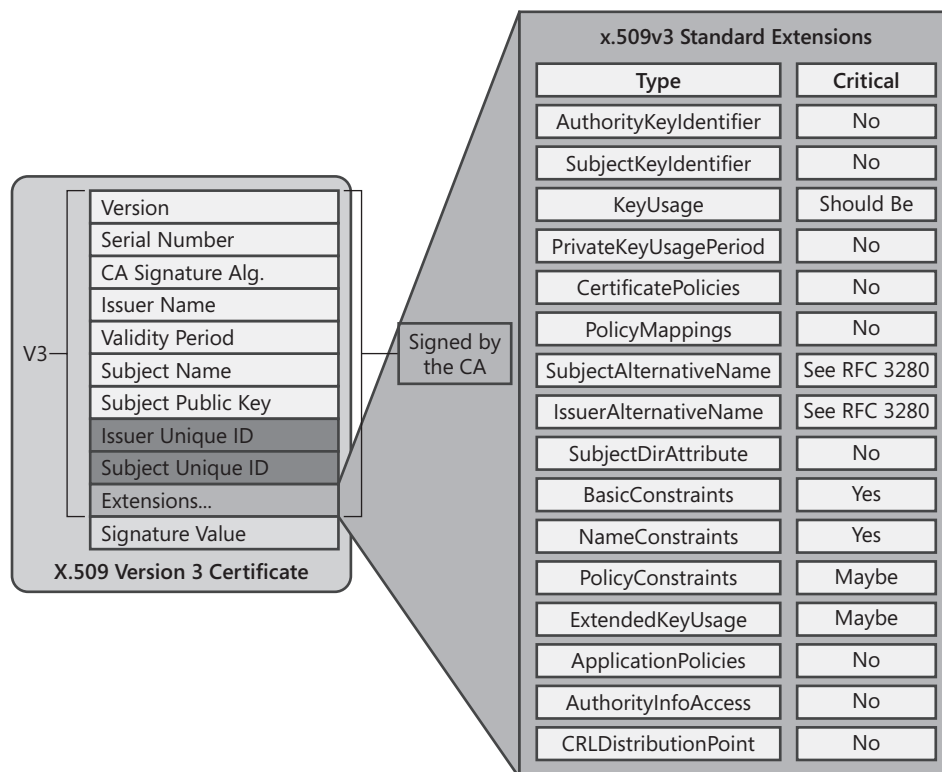
This additional level of matching allows a distinction between CA certificates when the CA renews a certificate. This method also allows for a distinction between CAs with the same subject name. (The likelihood of CA certificates with the same name increases when simple names are used—for example, *CN=Root CA* rather than *CN=Fabrikam Industries Inc. Corporate Root CA,O=Fabrikam,C=NL*.)

Although the addition of the Issuer Unique ID and Subject Unique ID aids in chain building, it's still possible for collisions to occur. A collision occurs when two certificates share the same Subject Name and Subject Unique Identifier fields.

Although the X.509 version 2 format improved on the version 1 format, the standard was not widely supported. In fact, RFC 3280, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile," recommends the omission of these X.509 version 2 fields.

## X.509 Version 3

Released in 1996, the X.509 version 3 format introduced **extensions** to address the problems associated with matching the Issuer Unique ID with Subject Unique ID, as well as other certificate-validation issues. An X.509 version 3 certificate can contain one or more certificate extensions. (See Figure 2-3.)



**Figure 2-3** The X.509 version 3 certificate fields

Each extension in an X.509 version 3 certificate is composed of three parts:

**Extension Identifier.**

- An object identifier (OID) that indicates the format and definitions of the extension.

**Criticality Flag.**

- An indicator that identifies whether the information in an extension is important. If an application cannot recognize the critical extension, the certificate cannot be accepted or used. If the criticality flag is not set, an application can use the certificate even when the application does not recognize the extension.

**Extension Value.**

- The value assigned to the extension. The value varies depending on the specific extension.

In an X.509 version 3 certificate, the following certificate extensions can exist:

**Authority Key Identifier.**

- This extension can contain one of two values. The value can be either
- The subject of the CA and serial number of the CA certificate that issued the current certificate.
- A hash of the public key of the CA certificate that issued the current certificate.

**Subject Key Identifier.**

- This extension contains a hash of the current certificate's public key.

The use of the Authority Key Identifier and Subject Key Identifier in certificate chaining and validation is described in Chapter 9, "Certificate Validation."

**Key Usage.**

- A CA, user, computer, network device, or service can have more than one certificate. The Key Usage extension defines the security services for which a certificate can be used. The options can be used in any combination and can include the following:

**Digital Signature.**

- The public key can be used to verify signatures. This key is also used for client authentication and data-origin validation.

**Non-Repudiation.**

- The public key can be used to validate the signer's identity, preventing a signer from denying that he/she signed a package.

**Key Encipherment.**

- The public key can be used for key transport for processes, such as symmetric key exchange. This Key Usage value is used when an RSA key is used for key management.

**Data Encipherment.**

- The public key can be used to directly encrypt data, rather than exchanging a symmetric key for data encryption.

**Key Agreement.**

- The public key can be used for key transport for processes such as symmetric key exchange. This value is used when a Diffie-Hellman key is used for key management.

**Key Cert Sign.**

- The public key can be used to verify a certificate's signature.

**CRL Sign.**

- The public key can be used to verify a CRL's signature.

**Encipher Only.**

- This value is used in conjunction with the Key Agreement Key Usage extensions. The resulting symmetric key can only be used for data encryption.

**Decipher Only.**

- This value is used in conjunction with the Key Agreement Key Usage extensions. The resulting symmetric key can only be used for data decryption.

**Private Key Usage Period.**

- This extension allows a different validity period to be defined for the private key of a key pair. The Private Key Usage Period can be set to a period shorter than the certificate's validity period. This gives the private key the ability to sign documents for a shorter period (say, one year), while the public key can be used to validate the signature for the certificate's entire five-year validity period.

**Certificate Policies.**

- This extension describes the policies and procedures used to validate a certificate's subject before the certificate is issued. Certificate policies are represented by OIDs. Optionally, a certificate policy can include a policy qualifier, which is typically a URL that describes, in text, the policies and procedures.

**Policy Mappings.**

- This extension allows for policy-information translation between two organizations. For example, imagine that one organization defines a certificate policy named Management Signing, which is in certificates used for signing for large purchase orders. Another organization can have a certificate policy named Large Orders, which also is used to sign large purchase orders. Policy mapping allows the two certificate policies to be deemed equivalent.

Policy mapping typically requires that the participating organizations' legal departments inspect each certificate policy. The policies can be deemed equivalent only after the legal departments are satisfied.

**Subject Alternative Name.**

- This extension provides a list of alternate names for the certificate's subject. While the subject can include the subject name in an X.500 distinguished name format, the Subject Alternative Name allows for other representations, such as a User Principal Name (UPN), e-mail address, IP address, or DNS name.



**Issuer Alternative Name.**

- This extension provides a list of alternate names for the issuing CA. Though it is not typically implemented, the Issuer Alternative Name extension can contain the e-mail name associated with a CA.

The Subject Alternative Name and Issuer Alternative Name extensions can be either critical or noncritical. RFC 3280 defines that if the Subject field is not empty, these extensions can be marked noncritical. If the Subject field is empty, these extensions must be marked critical to allow applications to inspect the names formats.

**Subject Dir Attribute.**

- This extension can include any attributes from an organization's X.500 or Lightweight Directory Access Protocol (LDAP) directory. For example, the country attribute from a directory can be included in the Subject Dir Attribute extension. This extension can contain multiple attributes from the organization's directory. For each attribute, the OID and its corresponding value must be included.

**Basic Constraints.**

- This extension allows a certificate to designate whether the certificate is issued to a CA or to a user, computer, network device, or service. Also, the Basic Constraints extension includes a path length constraint, which limits how many subordinate CAs can exist below a specific CA's issued certificate.

**Name Constraints.**

- This extension allows an organization to designate which name spaces are allowed or disallowed in a CA-issued certificate. A separate name constraint must be defined for each name-space format used in certificates. For example, separate constraints are required for LDAP names versus e-mail names.

**PolicyConstraints.**

- This extension can be included in CA certificates. The extension can prohibit policy mapping between CAs or require that each certificate in a certificate chain includes an explicit certificate policy OID.

**Enhanced Key Usage.**

- This extension indicates how a certificate's public key can be used. These are beyond the general purposes defined in the Key Usage extension. For example, OIDs exist for Client Authentication (1.3.6.1.5.5.7.3.2), Server Authentication (1.3.6.1.5.5.7.3.1), and Secure E-mail (1.3.6.1.5.5.7.3.4). When a certificate is presented to an application, an application can require the presence of an EnhancedKeyUsage OID specific to that application.

Enhanced Key Usage OIDs are also used when defining qualified subordination constraints. These constraints are discussed in Chapter 13, "Creating Trust Between Organizations."

**CRL Distribution Points.**

- This extension contains one or more URLs where the issuing CA's base CRL is published. If revocation checking is enabled, an application will use the URL to retrieve an updated version of the CRL. URLs can use Hypertext Transfer Protocol (HTTP), File Transfer Protocol (FTP), or LDAP.

**Authority Information Access.**

- This extension contains one or more URLs where the issuing CA's certificate is published. An application uses the URL when building a certificate chain to retrieve the CA certificate if it does not exist in the application's certificate cache.

**Inhibit Any Policy.**

- This extension is included in a CA certificate to inhibit the use of the All Issuance Policies OID (2.5.29.32.0) in subordinate CA certificates. This extension prevents the All Issuance Policies OID from being considered a match to a specific certificate policy OID in a subordinate CA certificate. The value of this extension defines the number of certificates that can appear below the CA certificate before the All Issuance Policies OID is not recognized.

**Freshest CRL.**

- This extension contains one or more URLs where the issuing CA's delta CRL is published. The delta CRL contains only the certificates revoked since the last base CRL was published. If revocation checking is enabled, an application will use the URL to retrieve an updated version of the delta CRL. URLs can use the HTTP, LDAP, or FTP protocols.

The use of base CRLs and delta CRLs is discussed in [Chapter 9](#), "Certificate Validation."

**Subject Information Access.**

- This extension contains information on how to access additional details about the certificate's subject. If the certificate is a CA certificate, the information can include particulars about the certificate validation services or the CA policy. If the certificate is issued to a user, computer, network device, or service, the extension can contain information about the services offered by the certificate subject and how to access those services.

In addition to introducing the extensions listed here, the X.509 version 3 certificate's Version field changed to a value of 3 to indicate the version number.

## Certification Authorities

A CA is an essential component of the Microsoft PKI solution. In a Windows Server 2003 network, a CA is a Windows Server 2003 computer with Certificate Services installed. It performs the following tasks:

**Verifies the identity of a certificate requestor.**

- The CA must validate the requestor's identity before it can issue a certificate. Validation can range from ensuring that the requestor has the necessary permissions

to ask for a specific type of certificate to having a certificate manager perform a face-to-face interview with the certificate requestor.

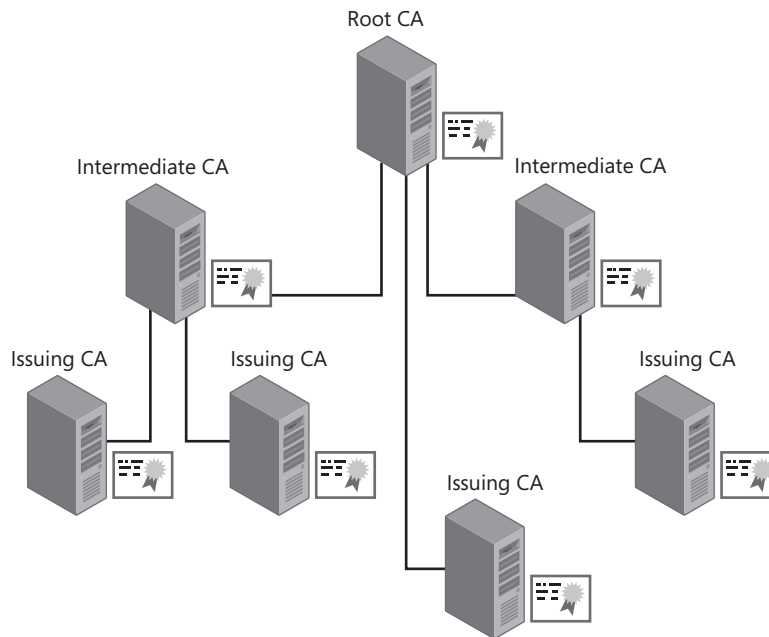
**Issues certificates to requestors.**

- After the requestor's identity is validated, the CA issues the requested type of certificate to the user, computer, network device, or service. The type of certificate requested determines the content of the issued certificate. For example, a Web server certificate request results in a certificate that can only be used by the Web server to setup Secure Sockets Layer (SSL) connections.

**Manages certificate revocation.**

- The CA publishes a CRL at regularly scheduled intervals. The CRL contains a list of serial numbers of certificates that are revoked and the reason codes for each revocation.

In an enterprise PKI, more than one CA is typically implemented. The CAs are organized into a CA hierarchy consisting of a single root CA and several other subordinate CAs, as shown in Figure 2-4.



**Figure 2-4** CA hierarchy roles

In Figure 2-4, the CAs are organized in a root CA hierarchy, which increases security and scalability of a CA hierarchy by allowing nonissuing CAs to be removed from the network. If the root CA and second-tier CAs in a root CA hierarchy are removed from the network, the offline CAs are protected from network-sourced attacks.

Do not assume that a root CA hierarchy always implements offline CAs. It is possible to deploy a root CA hierarchy without offline CAs, but it is not recommended because of security issues.

A root CA hierarchy allows the delegation of administration to different business units or divisions within an organization. Common-criteria role separation allows the designation of CA management roles at each CA in the hierarchy, giving different administration groups the ability to manage one CA in the CA hierarchy but not others.

The root CA hierarchy is supported by all leading commercial CA vendors, including RSA, Thawte, and VeriSign. The root CA hierarchy is also supported by most applications and network devices, allowing for interoperability with a variety of applications and network devices.

## Root CA

A root CA is the topmost CA in a CA hierarchy. In a PKI, the root CA acts as the trust point for certificates issued by CAs in the hierarchy. This means that if a certificate can be traced up through the CA hierarchy to a root CA that is trusted by a user, computer, network device, or service, the certificate is considered trusted.

A root CA is special in that its certificate is self-issued. This means that the certificate's Issuer Name and Subject Name fields contain the same distinguished name. The only way to validate whether a root certificate is valid is to include the root CA certificate in a trusted root store. The trusted root store contains the actual root CA certificate to designate that the certificate is trusted.

If a self-signed certificate is not included in the trusted root store, it is considered a nontrusted root CA. If revocation checking is enabled in an application, a certificate that is chained to a nontrusted root CA is considered nontrusted.

The root CA can issue certificates to other CAs or to users, computers, network devices, or services on the network. When the root CA issues a certificate to another network entity, the root CA certificate signs the certificate with its private key to prevent content modification and to indicate that the certificate was issued by the root CA.

Typically, the root CA only issues certificates to other CAs, not to users, computers, network devices, or services on the network.

## Intermediate CA

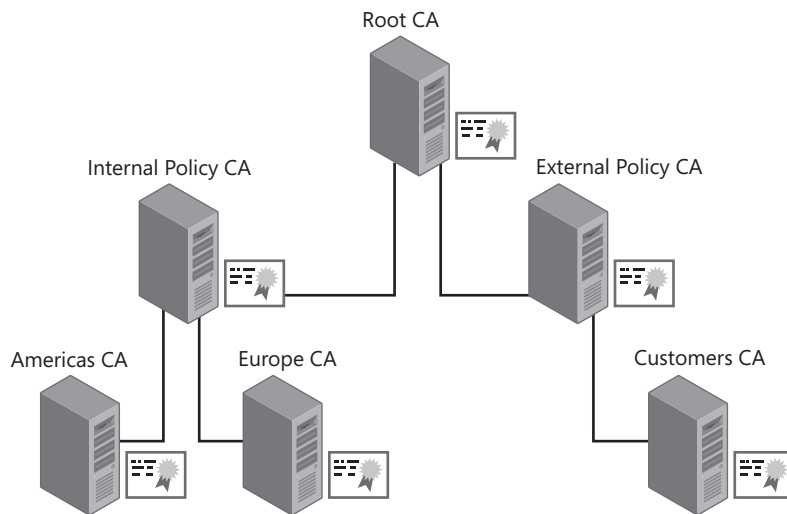
An *intermediate CA* is a CA that is subordinate to another CA and issues certificates to other CAs in the CA hierarchy. The intermediate CA can exist at any level in the CA hierarchy, except at the root CA level.

The CA that issues a certificate to another CA is often referred to as a *parent CA*. For example, a root CA that issues a certificate to an intermediate CA is referenced as the parent CA to the intermediate CA. The intermediate CA is also referred to as a *subordinate CA*, as it is directly subordinate to the parent CA in the hierarchy.

## Policy CA

A special category of intermediate CA is a policy CA. A *policy CA* describes the policies and procedures an organization implements to validate certificate-holder identity and secure the CAs in the CA hierarchy. A policy CA only issues certificates to other CAs in the hierarchy. It is assumed that all CAs that are subordinate to a policy CA, whether directly subordinate or two or more levels below the policy CA, enforce the policies and procedures defined at the policy CA.

If an organization must implement multiple policies and procedures when issuing certificates, multiple policy CAs must exist in the CA hierarchy. (See Figure 2-5.)



**Figure 2-5** Policy CA example

In this example, two policy CAs exist in the CA hierarchy. The Internal Policy CA defines the policies and procedures used to validate the identity of certificates issued to employees. The two issuing CAs (Americas CA and Europe CA), which are directly subordinate to the Internal Policy CA, must enforce the policies and procedures defined by the Internal Policy CA.

The External Policy CA defines the policies and procedures used to validate identity and secure the process of issuing certificates to nonemployees. The Customers CA, as a subordinate CA to the External Policy CA, must enforce the policies and procedures defined by the External Policy CA.

More than one policy or procedure can be defined at a policy CA, but it is also valid to implement one policy CA for each policy or procedure applied by the organization.

## Issuing CA

An issuing CA issues certificates to users, computers, network devices, or services on the network. An issuing CA is typically located on the third tier of a CA hierarchy, but it can exist on the second level, as shown in Figure 2-4.

As mentioned, an issuing CA must enforce any policies and procedures defined by a policy CA that exists between the issuing CA and the root CA in the CA hierarchy.

## Certificate Revocation Lists

In some cases, a CA must revoke a certificate before the certificate's validity period expires. When a certificate is revoked, the CA includes the serial number of the certificate and the reason for the revocation in the CRL.

### Types of CRLs

Windows Server 2003 supports the issuance of two types of CRLs: base CRLs and delta CRLs.

Windows Server 2003 does not support the issuance of indirect (or partitioned) CRLs.

A *base CRL* contains the serial numbers of all certificates revoked on a CA, as well as the reason for each revocation specific to a given private key used by the CA. The base CRL contains all certificates signed by a CA's specific private key. If a CA's certificate is renewed with a new key pair, a new CRL is generated that includes only revoked certificates signed with the CA's new private key.

A *delta CRL* contains only the serial numbers and revocation reasons for certificates revoked since the last base CRL was published. A delta CRL is implemented to provide more timely revocation information from a CA and to decrease the amount of data downloaded when retrieving a CRL. When a new base CRL is published, the revoked certificates in the delta CRL are rolled into the base CRL. The next delta CRL will only contain certificates revoked since the new base CRL was published.

The delta CRL is much smaller than a base CRL because only the most recent revocations are included. The base CRL, which contains all revoked certificates, can be downloaded less frequently.

If you implement delta CRLs, you must still download the base CRL. It is the combination of the base CRL and the delta CRL that provides the complete information on all revoked certificates.

### Revocation Reasons

When a certificate is revoked, the CRL entry can contain further information about the revocation. The reason codes can include:

#### Key Compromise.

- The private key associated with the certificate has been stolen or otherwise acquired by an unauthorized person, such as when a computer is stolen or a smart card is lost.

**CA Compromise.**

- The private key of a CA has been compromised. This can occur when the computer running Certificate Services or the physical device that stores the CA's private key is stolen. If a CA's certificate is revoked, every certificate issued by the CA is also considered revoked because the CA that issued the certificates is no longer considered trustworthy.

**Affiliation Changed.**

- The subject of the certificate, typically a user, is no longer affiliated with an organization.

**Superseded.**

- The revoked certificate has been replaced by a new certificate. This can occur because of changes in the extensions in a certificate or the certificate's subject name changes.

**Cessation Of Operation.**

- The certificate's subject has been decommissioned. This can take place when a Web server is replaced by a new Web server with a new name. Likewise, this can occur when a merger takes place and the previous DNS name is decommissioned, requiring replacement of all Web server certificates.

**Certificate Hold.**

- A revocation where a certificate is determined to be temporarily revoked. This can occur when an employee takes a leave of absence. The Certificate Hold reason is the only revocation reason that allows a certificate to be unrevoked.

Although Certificate Hold allows a certificate to be unrevoked, use of the Certificate Hold reason code is not recommended, as it can be difficult to determine if a certificate was valid at a specific time.

**Remove From CRL.**

- This reason is used when a certificate is unrevoked after being revoked with the Certificate Hold reason. This revocation reason is only used in delta CRLs to indicate that a certificate revoked in the base CRL is unrevoked in the delta CRL.

**Unspecified.**

- If a certificate is revoked without providing a revocation reason, the unspecified reason is automatically included in the CRL.

For more information about certificate revocation reason codes, see RFC 3280.

## Online Certificate Status Protocol (OCSP)

Windows Server 2008 introduces an alternative to certificate revocation lists to determine whether a certificate is revoked. Rather than a client download a base CRL or delta CRL, the client (OCSP client) sends an HTTP-based certificate status request to a server (referred to as an OCSP responder). The client determines the OCSP responder's URL by inspecting the certificate's Authority Information Access URL. If the URL contains an OCSP responder

URL and the client support OCSP, the client can proceed with sending an OCSP request to the OCSP responder.

The responder communicates with the certification authority that issued the queried certificate to determine the revocation status and returns a digitally signed response indicating the certificate's status. The OCSP responder can communicate directly with the certification authority or inspect the CRLs issued by the CA to determine the revocation status of the requested certificate.

The advantage of OCSP is that the amount of data in the request and response is a fixed size. The number of certificates actually revoked by the certification authority does not affect the size of the OCSP responder's response. Additionally, the OCSP responder typically provides more up-to-date revocation information to the OCSP client.

The biggest issue faced when deploying OCSP is scalability of the OCSP responder. High availability is a must, requiring multiple servers in a Windows Load Balancing Services (WLBS) cluster. The nodes in the cluster are often dispersed to major network hubs to allow timely responses to the OCSP clients.

## OCSP Client

Windows Vista and Windows Server 2008 support the use of OCSP for certificate revocation status determination. The OCSP client meets RFC 2560 - X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP but also implements the recommendations in standards track RFC 5019 - The Lightweight Online Certificate Status Protocol (OCSP) Profile for High-Volume Environments for optimization of OCSP for high-volume scenarios.

---

### What does RFC 5019 Support Add to OCSP?

The big reason for drafting RFC 5019 was to add functionality and performance to OCSP. Some of the major differences between the full profile (RFC 2560) and RFC 5019 include:

- The Lightweight OCSP Profile supports both the Hypertext Transfer Protocol (HTTP) and Secure Hypertext Transfer Protocol (HTTPS).
- Lightweight OCSP Profile responses must specify `notBefore` and `notAfter` dates, which are not required in the full profile.
- Signed requests are not supported in the Lightweight OCSP Profile. The client cannot create a signed request; if a signed request, which can be created by third-party OCSP clients, is sent to the Online Responder an "Unauthorized" response is returned.
- With the Lightweight OCSP Profile, `nonce` is not supported in the request and ignored in the response. However, the Online Responder supports the `nonce` extension and will return a response that includes the `nonce` extension if configured to do so.

*Ryan Hurst*

*Co-Author of RFC 5019*

---



## Online Responder Service

The Online Responder is a Microsoft Windows service (ocspsvc.exe) that runs on the OCSP server with Network Service privileges. The following operations are performed by the Online Responder service:

- Manages OCSP configuration. The Online Responder service attributes that can be configured include public interfaces, access control settings, audit settings, and Web proxy cache settings. The settings are stored in the registry of the OCSP server under HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\OCSPSvc\Responder.
- Retrieves and caches revocation information. The Online Responder can retrieve and cache revocation information (such as base CRLs and delta CRLs) for future responses.
- Signs OCSP responses. When the OCSP server responds to an OCSP request, the Online Responder service signs the response with a pre-defined signing key.
- Audits configuration changes. You can audit any changes to the configuration of the Online Responder. The auditing configuration meets Common Criteria requirements for auditing.

## Case Study: Inspecting an X.509 Certificate

In this case study, you will examine a sample certificate and answer questions related to the fields and extensions included in the certificate.

### Opening the Certificate File

Use the following procedure to open the sample certificate file on the compact disc that accompanies this book.

1. Insert the companion media in your CD-ROM drive.
2. Open Windows Explorer.
3. Open the folder CD:\Case Studies\Chapter2\
4. In the CD:\Case Studies\Chapter2 folder, double-click Samplecertificate.cer.
5. In the Certificate dialog box, click the Details tab.
6. From the resource materials for this chapter, open the Samplecertificate.cer file.

### Case Study Questions

1. What version is the certificate?

**The certificate is an X.509 version 3 certificate. You can verify this by viewing the Version field on the Details tab.**

2. What is the name of the issuing CA?

**The name of the issuing CA is CN=adatumCA,DC=adatum,DC=msft. You can verify this by viewing the Issuer field on the Details tab.**

3. What is the subject name of the certificate?

**The subject name of the certificate is CN = SCUser1, OU = Module09. OU = Labs, DC = adatum, DC = msft. You can verify this by viewing the Issuer field on the Details tab.**

4. Are any other names included in the certificate for the subject?

**The Subject Alternative Name extension contains an additional name for the subject. The name is a user principal name, SCUser1@ADATUM.msft.**

5. What is the length of the public key associated with the certificate?

**The public key length is 1024 bits. You can verify this by viewing the Public Key field on the Details tab.**

6. What other X.509 extensions are included in the sample certificate?

**On the Details tab, the following X.509 version 3 extensions are included: Key Usage, Application Policies, Certificate Policies, Enhanced Key Usage, Subject Key Identifier, Authority Key Identifier, CRL Distribution Points, Authority Information Access, and Subject Alternative Name.**

7. What extensions must you inspect to determine what forms of revocation checking are supported by the CA that issued the X.509 certificate?

**You must inspect both the Authority Information Access and CRL Distribution Point extensions to determine what forms of revocation checking are supported by the CA that issued the X.509 certificate. If the CA supports OCSP, the URL of the OCSP Responder is included in the Authority Information Access extension. If the CA supports CRLs, the URLs for CRL Distribution Points are included in the CRL Distribution Point extension.**

8. What forms of revocation checking are supported by the CA that issued the X.509 certificate?

**The CA only supports CRLs. The Authority Information Access extension does not contain any URLs referencing an OCSP responder.**

9. Where is the CRL published when revocation checking is performed against the certificate?

**On the Details tab, two URLs are included in the CRL Distribution Points extension indicating where the CRL is published:  
*ldap://CN=adatumCA,CN=VANCOUVER,CN=CDP,CN=Public%20Key%20Services,CN=Services,CN=Configuration,DC=adatum,DC=msft?certificateRevocationList?base?objectClass=cRLDistributionPoint* and  
*http://vancouver.adatum.msft/CertEnroll/adatumCA.crl.***

## Additional Information

- Microsoft Official Curriculum, course 2821: "Designing and Managing a Windows® Public Key Infrastructure" ([www.microsoft.com/traincert/syllabi/2821afinal.asp](http://www.microsoft.com/traincert/syllabi/2821afinal.asp))
- RFC 3280 - Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile (<http://www.faqs.org/rfcs/rfc3280.html>)
- RFC 2560 - X.509 Internet Public Key Infrastructure Online Certificate Status Protocol – OCSP (<http://www.faqs.org/rfcs/rfc2560.html>)
- RFC 5019 - The Lightweight Online Certificate Status Protocol (OCSP) Profile for High-Volume Environments (<http://tools.ietf.org/html/rfc5019>)