Chapter

4

Public Key Technology in Windows 2000

The Windows 2000 operating system has a built-in public key infrastructure (PKI) to address the business needs of enterprises that want to conduct e-commerce over the Internet. The built-in PKI provides a distributed authentication model that scales to the Internet and that interfaces with existing PKI trust infrastructures, enabling large-scale deployment of e-commerce applications. Furthermore, an enterprise can leverage the built-in PKI to enhance the security of its internal networks by using, for example, smart cards instead of passwords for domain network log-on.

We start by presenting a list of Windows 2000 applications that use public key technology to address their security needs. We then discuss the Windows 2000 public key security architecture and provide basic information on the Windows 2000 PKI. Finally, we turn our attention to the interoperability issues and examine the various levels of interoperability between Windows 2000 PKI and a third-party PKI.

Public Key Security

Windows 2000 leverages public key technology to address the security needs of a wide range of real-world business-to-consumer and business-to-business applications. This section presents the major applications that have an underlying public key security.

Secure E-Commerce: TLS/SSL

The Internet has already crossed the chasm between a publishing platform and a platform to conduct on-line business. Shopping malls and merchants have set up secure Web sites to extend their businesses to on-line consumers and to receive payments. The secure Web sites enable consumers to verify the identity of merchants and to ensure the privacy of their transactions and payment information.

Windows 2000 provides an infrastructure to enable business-to-consumer e-commerce. The support for Secure Socket Layer (SSL) 3.0 [FRIE96] and Transport

Layer Security (TLS) 1.0 protocols [DIER99], public key certificates, and embedded trust points in browsers are the key cornerstones of this infrastructure. The TLS/SSL protocols provide security over public networks and prevent communications eavesdropping, tampering, and forgery. Client/server applications use the TLS handshake protocol to authenticate each other and to negotiate an encryption algorithm and cryptographic keys before an application starts transmitting data. The handshake protocol uses public key cryptography, such as RSA [RIVE78] or DSS [DSS94], to authenticate peers and to negotiate a shared secret. Public key certificates provide evidence for the identity of merchants; consumers use their own local policies to decide how much trust to place in these certificates.

Once a channel is authenticated, TLS uses symmetric cryptography, such as DES [DES83] or RC4, to encrypt the application data in the negotiated shared secret prior to transmission over the network. Message transmission includes a message integrity check, using a keyed message authentication code (MAC) [KRAW97], computed by applying a secure hash function, such as SHA [SHA94] or MD5 [RIVE92]. Encryption of application data ensures the privacy of communications and payment information with a Web server, whereas message integrity checks prevent communications tampering and forgery.

Supporting Distributed Business Partners: TLS/SSL Client-Side Authentication

The Internet is undoubtedly the ultimate platform for distributed computing. The general public uses the Internet to access information, to view catalog information, and to place orders. Company employees who are on the road or working from home connect to their favorite Internet service providers and access their company's intranet to carry out their tasks. Similarly, a company's business partners use the Internet to access privileged resources from the company's extranet and to perform a variety of business-to-business processes, such as supply chain management and customer relations management.

Supporting on-line business-to-business relationships poses a unique challenge: the need for distributed authentication. An enterprise must be able to reliably authenticate its distributed partners to determine and to enforce their access rights to its internal resources. The authentication mechanism must scale to thousands of partners—millions of consumers for business-to-consumer applications—and must be flexible; it must be administratively easy to add a new business partner or to remove a partner no longer authorized to use the extranet.

Windows 2000 leverages public key technology to offer a flexible solution for distributed authentication. An enterprise can issue client certificates for its business partners or consumers and use Windows 2000 PKI to authenticate partners, based on their certificates. The authentication hinges on the client-side authentication in

TLS, public key certificates, and trust points in a Web server. During the TLS handshake protocol, a Web server can ask a client for its certificate and confirm the client's identity, based on the submitted certificate and the Web server's trust policy. The protocol allows a browser to display a suitable list of available certificates to a client; browsers enhanced with additional software can further customize this list and provide branding information.[1]

After a Web server verifies a client certificate, Windows 2000 provides a mapping mechanism to relate the external identity of a distributed user to an internal enterprise identity. Windows NT 4.0 supports the mapping within the Internet Information Services (IIS); Windows 2000 provides an alternative approach by defining the mappings within Active Directory. The mapping can be either many-to-one, associating many clients to one Windows account, or one-to-one, relating one client to one Windows account. Windows 2000 provides a great deal of flexibility for setting up the mapping relationships between client public key certificates and Windows accounts, such as using the certificate issuer or subject fields as mapping parameters.

Windows 2000 uses the enterprise identity of an external client for updating account information and generating audit trails. Equally important is enforcing access-control rules and ensuring that a distributed partner accesses only the intended resources. When mapping an external user to a Windows account, Windows 2000 uses the access rights of the mapped account to determine and to enforce access rights. An enterprise can set up an account for each distributed partner with the proper access rights to its extranet; Windows 2000 provides built-in operating system support to enforce the rights.

Strong Network Authentication: Smart Cards

Passwords have traditionally been a weak link in the overall security of an authentication system [FEGH98]. Passwords have poor random qualities because humans need to be able to memorize them. Users typically need to remember a number of passwords to access various systems and tend to forget their passwords, requiring an administrative process to reissue new passwords, which further weakens the overall security of a system.

Windows 2000 supports smart cards for strong, interactive network authentication. Smart cards hold cryptographic public key–based keys that have much better random qualities than do passwords. Users interactively log on to their domain accounts by proving that they are in possession of the private keys corresponding to their public key certificates. Windows 2000 implements the required public key

1. VeriSign offers a product called Personal Trust Agent (PTA) that improves the user experience and provides branding for client certificates.

extensions to Kerberos to enable smart card log-on. Furthermore, Windows 2000 PKI has the necessary machinery to issue certificates in smart cards for network users. See Chapter 2 for more information on Windows 2000 smart card interactive log-on.

Distributing Authenticated Code: Authenticode 2.0

The Internet provides an extremely effective platform for distributing software. Many Web sites have content containing downloadable code, such as ActiveX controls, Java applets, or scripts, that transports to browsers during Web surfing. Once downloaded, the code runs on the client computers and performs tasks ranging from simple error checking on HTML (Hypertext Markup Language) forms to such sensitive operations as reading personal files. Downloadable code adds programming logic to digital content, enhances the functionality of the browser, and improves the user experience. Unfortunately, rogue Web sites can use the same distribution channel to download harmful code to client computers for fraudulent purposes. Furthermore, attackers can infect downloadable code with a virus while in transit from a legitimate Web server to end users' desktops.

Authenticode 2.0 provides accountability for downloadable code and ensures the integrity of code while in transit. Authenticode uses public key certificates issued for software publishing to create a digital signature over an executable program, a cabinet file, a digital thumbprint, an ActiveX control, a dynamic link library (DLL), or a certificate trust list (CTL). The signature binds the code to the identity of its publisher; the software publishing certificate vouches for the identity of the publisher and creates accountability. When a Web surfer downloads digital content that contains signed code, a browser interrupts[2] the download process and prompts the user for approval. Trust in the certification authority that has issued the software publishing certificate, the software publisher, and other local policy trust decisions determine whether the user approves the signed code. See [FEGH98] for an overview of the Authenticode technology.

Laptop and Desktop File System Security: EFS

The Windows NT file system (NTFS) protects sensitive files against improper access but is helpless to prevent an attacker from running another operating system, such as UNIX or MS-DOS, to inspect NTFS-based files on disk structures. An attacker can boot another operating system from the floppy when a computer boots or may physically remove a hard disk and install it in a computer with a different operating

2. Users can configure their browsers to automate such trust decisions.

system. Tightening the physical security helps minimize such attacks, but such measures are not as effective against insider attacks and do not work when an employee carries around sensitive information on a laptop.

Data encryption provides the only safeguard against such attacks. A stolen laptop or hard disk is useless if the attacker cannot decipher encrypted files. Although a number of products in the marketplace offer application-level file encryption, they generally suffer from inherently weak password-derived keys for encryption, are not transparent, and do not have recovery agents.

Windows 2000 provides a built-in data encryption service called **Encrypting File System** (**EFS**). EFS uses symmetric key cryptography for encryption and public key cryptography for securing the random symmetric keys. Encryption and decryption of files are transparent to end users and happen seamlessly when data travels to and from disk structures. EFS supports file sharing among any number of users by keeping a copy of a random symmetric key encrypted in the public key of each user. Built-in data recovery agents allow an enterprise to enforce its local policy on EFS, such as recovering encrypted files when employees leave or when they lose their private keys. Refer to [MICR00D] for an in-depth discussion of EFS.

Secure E-Mail: S/MIME

The use of e-mail for business-to-consumer transactions has already taken off as a replacement for regular mail. Businesses now use e-mail to inform consumers about their promotions, send monthly billing statements, confirm stock trades, and so on. Conventional Internet e-mail, however, does not provide the same quality of service that regular mail provides. E-mail is, for example, vulnerable to eavesdropping and counterfeiting. Secure e-mail, however, provides many of the protections that people associate with regular mail, providing message origin authentication, message integrity, nonrepudiation of origin, and message confidentiality. Secure e-mail furnishes **writer-to-reader security**, which protects an e-mail from the moment it leaves a sender's mailing tool until it arrives at a recipient's mailing tool.

Windows 2000 supports the S/MIME (Secure Multipurpose Internet Mail Exchange) protocol for securing e-mail messages in the Internet. S/MIME leverages symmetric key cryptography for confidentiality, public key cryptography for authentication and nonrepudiation, and a formal public key infrastructure for accountability. The Windows 2000 built-in PKI provides the required machinery to implement S/MIME in an enterprise. The flexibility of the Windows 2000 PKI allows an enterprise to chain an internal trust point to an external, commercial trust point, in order to extend the secure e-mail protections beyond its internal boundary. An enterprise also has the flexibility to outsource the entire management for its S/MIME PKI to a third-party trust provider, such as VeriSign. Chapter 5 and Chapter 9 discuss such integration considerations with third-party trust providers and external trust infrastructures in detail.

Network-Level Secure Communications: IPsec

Securing network traffic at the IP layer provides transparency and end-to-end security. Applications and higher-layer protocols, such as TCP or UDP, can transparently leverage the IP-layer security services without requiring any code changes. The provided end-to-end security services protect packets from the moment they leave a source IP node until they arrive at a destination IP node. In contrast, security services at a layer above the IP layer do not have the transparency property; security services at a layer below the IP layer do not have the end-to-end property.

IP Security (**IPsec**) lays a security architecture for the Internet Protocol and provides high-quality, cryptographically based security services for authentication, integrity, confidentiality, and access control. IPsec-enabled systems select the security features they need and communicate securely over insecure networks with other IPsec-enabled systems. IPsec secures IP packets at the network level according to the security policy of a communicating IP node before forwarding them to the network interface layer for transmission; the intended receiving IP node verifies the packets according to the established security associations and rejects packets that do not have the expected level of security.

Windows 2000 provides a built-in implementation of the IPsec security protocol and its associated key management protocols. Windows 2000 default IPsec policies govern how clients and servers engage in secure communications; network administrators can create custom policies to enforce their local business rules. Windows 2000 supports router-router virtual private networks (VPNs) based on IPsec and remote access virtual private networks based on L2TP/IPsec. We will provide detailed coverage of IPsec and virtual private networks in Part III.

Public Key Security Architecture

Windows 2000 provides a public key security architecture to support a wide range of applications that require distributed, scalable authentication. This architecture does not replace the default Kerberos authentication protocol but rather addresses the needs of an enterprise to expand its business to the Internet. The components of this architecture seamlessly integrate with the operating system, instead of being an adjunct to it, and effectively interface with external trust infrastructures. See Figure 4-1 for an architectural diagram.

Microsoft CryptoAPI is the cornerstone of this architecture, providing the machinery to build authentication, integrity, and privacy security services into applications and supporting functions to manage public key certificates, CRLs, and PKCS messages (refer to Chapter 3 to an explanation of PKCS). CryptoAPI interfaces with installable cryptographic service providers (CSPs), discussed in the next section, to implement its functions, either in software or in hardware.

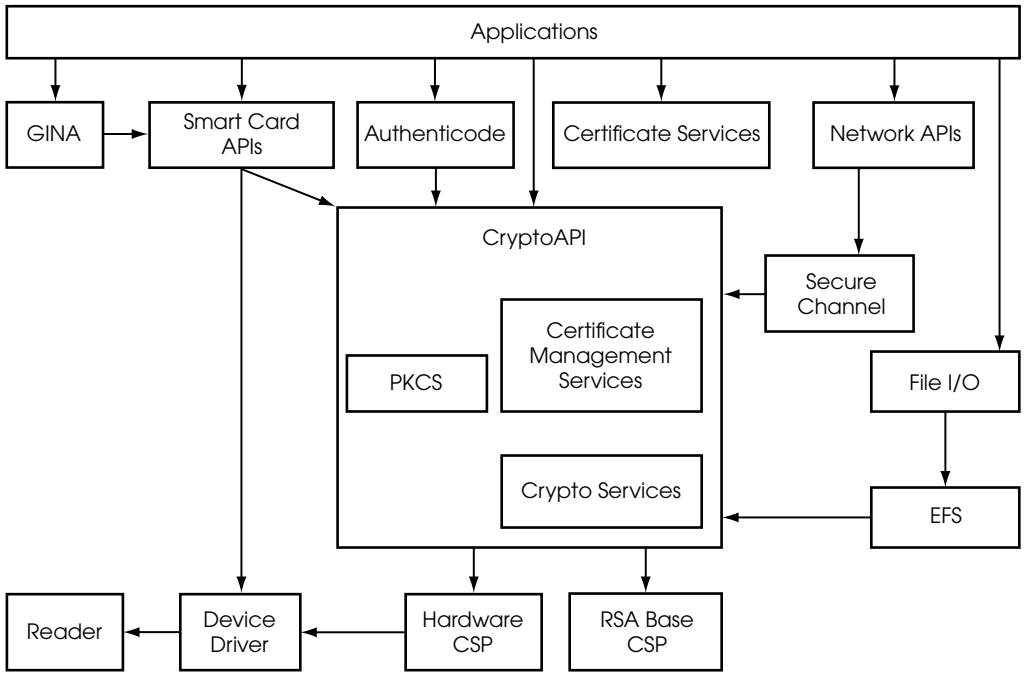

Figure 4-1 Windows 2000 public key security architecture

Other technologies layer on top of CryptoAPI. Authenticode provides accountability for distributing code in the Internet. Certificate Services functions as an in-house certification authority and provides certificate life-cycle management functions. Secure channel (SChannel) supports distributed authentication for the Internet, using the industry-standard TLS/SSL protocols. EFS provides transparent privacy and recovery services for file systems. A component of Microsoft Exchange, the Exchange Key Management Service (KMS) provides archiving and retrieval facilities for e-mail encryption keys.[3] Finally, general-purpose smart card interfaces provide an application-independent framework to integrate smart cards with applications.

At the very top are a myriad of PKI-enabled applications. Internet Explorer, Internet Information Services, Outlook, Outlook Express, and Microsoft Money are examples. In addition, a number of third-party applications integrate with the Windows 2000 PKI.

3. A future version of Windows will subsume KMS into the operating system.

CryptoAPI

Microsoft Cryptographic Application Programming Interface (**CryptoAPI**) enables programmers to incorporate authentication, integrity, and privacy security services into their applications. CryptoAPI supports a wide range of security mechanisms. Context functions connect to a specific CSP by name or connect to a CSP that can provide a needed class of functionality. Key generation functions generate and store cryptographic keys; key exchange functions exchange and transmit keys. Certificate encode functions, decode functions, and store functions manage public key certificates. Other cryptographic functions encrypt data, decrypt data, sign messages, and verify the authenticity of signatures on messages.

The CryptoAPI provides a framework for developing portable applications. The interface gives programmers a great deal of control and flexibility over the incorporation of the required security protections into their applications. Programmers use the interface, for example, to choose the encryption algorithm and key length providing the required quality of privacy. The CryptoAPI, however, carefully restricts application access to the cryptographic internals. Applications use the CryptoAPI functions without knowing anything about the underlying implementation or being able to specify the details of cryptographic operations. As a result, CryptoAPI-based applications can run in a variety of environments with differing configurations of cryptographic service providers. Application development with CryptoAPI parallels applications that use a graphics library without knowing anything about the particular graphics hardware configuration.

The CryptoAPI enhances security by preventing an application from accidentally divulging keying material or choosing keying material from a poor random source. CryptoAPI uses underlying CSPs for the generation and management of all keys. Applications cannot directly access and expose their keys but instead reference them only through opaque handles.

Applications built on top of CryptoAPI can adapt to a system's underlying authentication capabilities. The CSP performs user authentication and manages the credential data; CryptoAPI applications do not handle user credentials or other user authentication data. As a result, applications can adjust to future CSPs with advanced authentication capabilities, such as biometrics, without needing to change their authentication model.

Cryptographic Service Providers

A **cryptographic service provider** (**CSP**) contains implementations of cryptographic standards and algorithms. CryptoAPI functions use CSPs to generate and to store cryptographic keys, to manage public key certificates, to encrypt and to decrypt data, to sign messages, and to verify signatures. At a minimum, a CSP consists of a

dynamic link library that implements the CryptoAPI functions and an associated signature file. Ideally, applications based on CryptoAPI are independent of a particular CSP and run with a variety of CSPs. In reality, however, an application may have specific security requirements that require a particular CSP.

Depending on the capabilities of the underlying CSP, an application can run in a variety of security contexts without any modification. Different CSPs provide different levels of protection for cryptographic keys. Some CSPs use hardware devices, such as smart cards, to ensure that signature keys never leave the perimeter of the device, whereas other CSPs provide only software implementations of the CryptoAPI functions. CSPs also differ in their encryption strength. Some provide low-strength 512-bit RSA public keys and 40-bit encryption; others supply 1,024-bit RSA keys and 128-bit encryption.

Each CSP uses a **key database** to store cryptographic keys. A key database contains one or more **key containers**, each of which stores all the public/private key pairs belonging to a specific CryptoAPI client. CSPs store key containers in a permanent storage, such as a disk file, under a registry key, or in a hardware device. Each key container has a unique name, typically the login name of the user who owns the keys. Key stores, discussed in the next chapter, store certificates.

The Windows 2000 operating system bundles one CSP, the **Microsoft Base Cryptographic Provider**. This software-based CSP uses RSA signing keys of up to 16,384 bits for digital signatures and RSA key-exchange keys of up to 1,024 bits for exchanging symmetric key. It also implements the RC2, RC4, and RC5 encryption algorithms with a maximum key length of 56 bits and provides MD2, MD5, and SHA hashing algorithms. **Microsoft Enhanced Cryptographic Provider** provides stronger cryptographic capabilities. Refer to Table 4-1 in the section Cryptographic Algorithms and Key Lengths for further details on these two CSPs.

Certificate Services

Certificate Services, which plays a pivotal role in the overall public key security infrastructure of Windows 2000, functions as an in-house certification authority and provides certificate life-cycle management services to domain users. Certificate Services can automatically issue certificates to network computers when they boot and can publish the issued certificates into Active Directory.

The architecture of Certificate Services supports extensibility and programmability. An enterprise can replace default modules with custom modules to enforce its own local policy on certificate issuance, contents of certificate fields, certificate extensions, and external certificate storage requirements. The architecture can extend Certificate Services to support custom certificate extensions, custom certificate types, and custom transport protocols. Furthermore, an enterprise can use Certificate Services to set up a certificate hierarchy that matches its organizational structure.

Certificate Services accepts certificate enrollment requests in the standard PKCS #10 format and issues standard X.509 V3 certificates. Certificate Services is transport independent: A client application can use a variety of transport-layer protocols, such as HTTP, RPC, DCOM, e-mail, file disk, or custom transports, to submit an enrollment request and to receive a certificate.

Certificate Services issues certificates for various applications, including client authentication, Web server authentication, and secure e-mail, and provides revocation information for certificate-using applications through CRLs and CRL distribution points. Certificate Services supports hardware-based cryptography, such as issuing certificates on smart cards for domain log-on or other applications that require smart cards.

Certificate Services supports both enterprise and standalone certification authorities.[4] Enterprise CAs tightly integrate with Active Directory and automatically process enrollment requests, whereas standalone CAs typically queue enrollment requests for manual processing. Chapter 5 covers enterprise and standalone CAs in more detail.

Certificate Services consists of the server engine, the server database, an intermediary, a set of customizable modules, and administration tools. The flexibility to customize Certificate Services is perhaps the most notable aspect of the architecture. The server engine provides a number of Component Object Model (COM) interfaces. An enterprise dictates its own local policy for certificate approval, certificate fields, certificate extensions, and external certificate storage requirements by customizing the default policy and exit modules, which interact with the server engine through the COM interfaces. Furthermore, the server engine can encode complex, custom extensions through an extension-handler mechanism.

Each building block of Certificate Services plays a role during the certificate enrollment process. An intermediary application receives a certificate request from a client, formats it into a PKCS #10 certificate request, and submits it to the server engine. The server engine hands over the request to the policy module. This module either approves or denies the request, optionally adjusts the certificate subject distinguished name, potentially adjusts the certificate validity period, and sets optional certificate extensions. If the policy module has approved the request, the server

4. To avoid a possible naming confusion among many PKI products and services, we adopt the following conventions in this book. The terms Microsoft or Windows 2000 enterprise CA and Microsoft or Windows 2000 standalone CAs refer to Microsoft or Windows 2000 Certificate Services. We may omit Microsoft or Windows 2000 if suitable contextual information exits. The term third-party PKI product refers to a PKI product other than Microsoft PKI and its associated Certificate Services. The terms third-party commercial PKI service provider and third-party commercial CA service provider refer to third-party certification authorities that provide PKI outsourcing services.

engine builds a certificate to fulfill the original request and the adjustments made by the policy module and stores the certificate in its database. The policy module then delivers the issued certificate to the intermediary application, which passes the certificate back to the client for installation. Finally, the exit module receives a certificate issuance event notification from the server engine, if it has so requested, and performs further operations, such as publishing the certificate to Active Directory.

Server Engine

The server engine builds certificates and CRLs. It communicates with clients through an intermediary, receives certificate requests, generates certificates, and delivers them back to the intermediary. The server engine communicates with a customizable policy module before issuing a certificate and notifies a customizable exit module when issuing a certificate or a CRL.

Policy Modules

Policy modules allow an enterprise to enforce its local policy on certificate issuance. Most important, a policy module determines whether the server engine should fulfill a certificate enrollment request. Exactly how a policy module arrives at this critical decision is a matter of local policy and reflects an enterprise authentication model and practice statements. The policy module can, for example, approve a request if a user has provided appropriate Windows 2000 credentials, or it can consult a third-party authentication database.

Policy modules also enable an enterprise to enforce its local policy on the content of a certificate. A policy module can make certain changes to the certificate subject distinguished name, modify the requested validity period, and add custom extensions to the certificate.

A policy module is a customizable dynamic link library. Windows 2000 ships appropriate default policy modules for both the Certificate Services enterprise and standalone CAs. The enterprise CA default policy module automatically approves authorized certificate requests, adds a set of predefined extensions to certificates, and supports smart card certificates for domain log-on. The standalone CA default policy module queues enrollment requests for manual approval and does not support many of the features of the enterprise default policy module. An enterprise can provide its own custom policy module if the default modules are unacceptable; Microsoft, however, recommends the default enterprise policy module for enterprise CAs.

Exit Modules

Exit modules deal with the storage of certificates and certificate revocation lists (CRLs) in external repositories. When it issues a certificate or a CRL, the server

engine interacts with an exit module to enforce the local storage policy. Note that an exit module can only publish a certificate or a CRL into a repository but cannot make any changes to them.

An exit module is a customizable dynamic link library. Windows 2000 ships an enterprise certification authority default exit module that publishes certificates and CRLs into Active Directory. An enterprise can provide its own custom exit module if the default module is unacceptable. For an enterprise certification authority, however, an enterprise should always use the default exit module as a part of its custom exit module.

Extension Handlers

A policy module can encode domain-specific information in a certificate by adding X.509 V3 extension fields. For extensions that have date, long, or string data types, the policy module passes the required extensions and their types to the server engine, which ASN encodes the extensions and stores them in the certificate. If an extension has a type other than date, long, or string, the policy module must ASN-encode the extension before handing it over to the server engine.

An extension-handler module is a customizable dynamic link library that provides the necessary machinery to ASN-encode complex extensions. The policy module simply calls the appropriate extension-handler function to perform the ASN encoding before passing the extension to the server engine. Windows 2000 ships a default extension-handler module that encodes various complex extensions, such as alternative names, bit strings, CRL distribution points, date arrays, long arrays, and string arrays. An enterprise can write its own extension-handler module to encode custom extensions or other complex extensions.

Intermediaries

Intermediaries work as a glue between client applications and the server engine. An intermediary receives certificate enrollment requests from a client over a transport protocol, such as HTTPS, and submits them to the sever engine for processing. The intermediary locates the enterprise certification authority that should process the enrollment, submits the request, receives the issued certificate, and delivers the certificate back to the client.

Microsoft Internet Information Services (IIS) is the standard HTTP intermediary application for Certificate Services. IIS receives enrollment requests over the HTTP/HTTPS protocol from Web clients—browsers—and communicates with Certificate Services to fulfill the requests. An enterprise needs to develop intermediaries for clients that use other transport layer protocols, such as RPC or e-mail.

Public Key Infrastructure

The public key infrastructure in Windows 2000 [MICR00A, MICR00B] supplies public key capabilities to applications that use certificates for making automated identity assessments and trustworthiness decisions. Applications leverage the Window 2000 PKI to establish the quality of protection they need before they process a transaction or open a communications channel. This section covers some of the Windows 2000 PKI features.

Trust Models

This section discusses three models for controlling the flow of trust in a network. Refer to the section Structures among Multiple Certification Authorities in Chapter 3 and the section Models for CA Structures in Chapter 5 for related information.

The Rooted Hierarchical Trust Model

In a **rooted hierarchical trust model**, trust starts with one top-level root CA and flows down the hierarchy through a chain of subordinate CAs to end-entities. All CAs in a hierarchical PKI are subordinates to other, superior CAs, except for the top-level root CA, which has a self-signed certificate and does not have a superior CA. Members of a hierarchical PKI need to trust only the top-level root CA in order to trust all the other member CAs. Numerous PKI products and third-party commercial CA service providers—notably Microsoft and VeriSign [http://www.verisign.com]—support the rooted hierarchical trust model. Figure 4-2 shows a three-level hierarchical PKI.

A new CA can enter a hierarchical PKI through the process of subordination, in which an existing CA issues a certificate for the new CA. The subordination process is transparent and does not impact the users of the hierarchy; they can correctly process the certificates issued by the new CA without any configuration changes. Integrating an existing, foreign CA into a hierarchical PKI, however, is more problematic. Subordinating the foreign CA changes the relative point of trust for the users of the foreign PKI. They must now directly trust the root CA of the other PKI hierarchy instead of their own, which may be difficult to achieve in a peer-to-peer business relationship. The most effective way to integrate a foreign hierarchical PKI is for each member user to directly trust the foreign CA—in a secure manner—in addition to its own CA. Care must be taken to limit the use of the foreign CA to the intended purposes only, however. Windows 2000 provides certificate trust lists to disseminate such trust relationships; the section Certificate Trust Lists explains CTLs.

Rooted hierarchical trust models offer a scalable, easy-to-administer PKI because each CA serves a specific role in the hierarchy: as either a root CA or a subordinate CA.

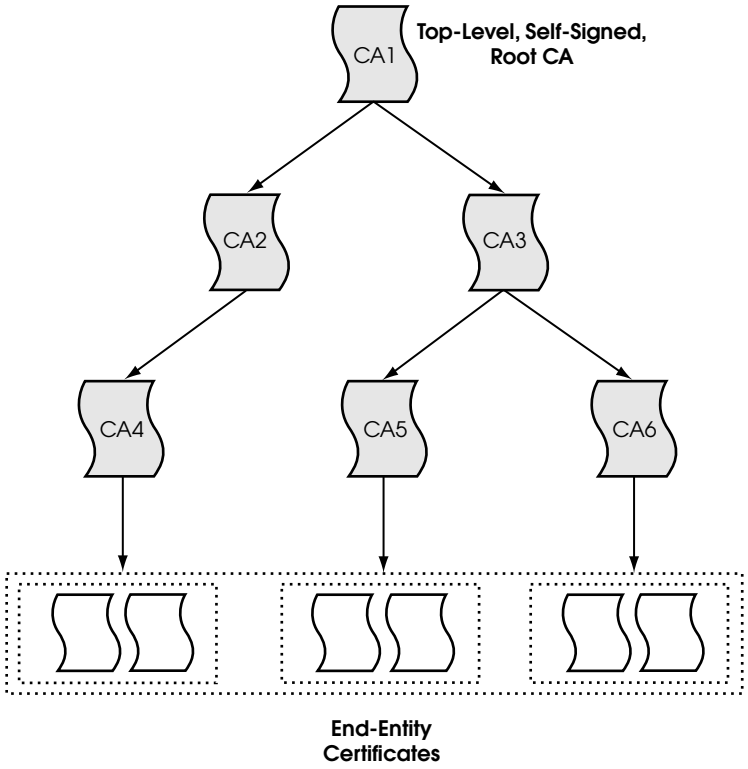

Figure 4-2 A rooted hierarchical PKI

Each member CA processes enrollment requests, issues certificates, and maintains revocation information for its own jurisdiction. Most important, client applications do not need to access a global repository of CA certificates to verify certificate chains. The self-signed root is the natural point of trust, making it straightforward to establish trust in a given certificate, especially when client applications submit complete certificate chains. Secure e-mail applications based on S/MIME and Web browsers belong to this category of applications.

The Network Trust Model

In a **network trust model**, all CAs are self-signed, and trust flows through the network via cross-certificates. An end-entity directly trusts the CA closest to it and trusts another CA if its direct CA has cross-certified the foreign CA. Because cross-certification creates a parent-child relationship between two CAs, a networked PKI is also a hierarchical PKI with the distinction that the self-signed root is a subordinate CA in the cross-certifying PKI. Entrust Technologies [http://www.entrust.com]

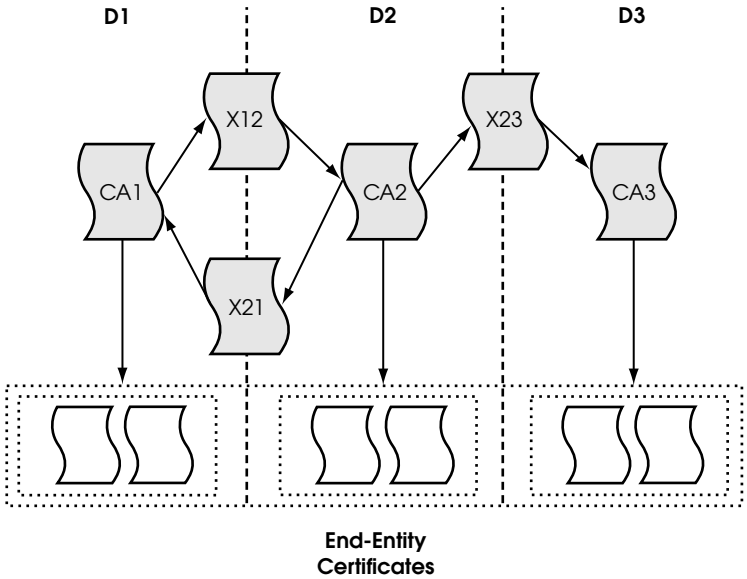

Figure 4-3 A networked PKI

is perhaps the most widely known PKI vendor that supports the network trust model.[5] Figure 4-3 shows a sample networked PKI.

The network of Figure 4-3 shows three PKI domains that network with one another through three cross-certificates. The certificate X12 represents a **cross-certificate** between CA1 and CA2; CA1 is the **cross-certifying CA**, whereas CA2 is the **cross-certified CA**. The certificate X21 plays the reverse role of the X12 certificate and allows CA2 to cross-certify CA1. The combination of X12 and X21 cross-certificates creates **a bilateral cross-certification** between domains D1 and D2. Domains D2 and D3, however, have established a **unilateral cross-certification**: CA2 has cross-certified CA3 but not vice versa. Bilateral cross-certification offers better interoperability behavior than does unilateral cross-certification. Members of domain D1 and D2 can exchange secure, authenticated e-mail, for example, whereas members of domain D3 can send authenticated e-mail to members in domain D2 but will have trouble receiving authenticated e-mail from them.

A new CA enters a networked PKI through the process of **cross-certification**, in which an existing CA issues a cross-certificate for the new CA. An existing CA leaves the network by revoking its cross-certificate. The cross-certification process is transparent and does not impact the users of the network, provided that they can retrieve

5. Entrust plans to support the hierarchical trust model in the 5.0 release of its products.

the cross-certificates from a global directory. The cross-certification process can also integrate an existing, foreign CA into a networked PKI without changing the relative point of trust for either PKI. Compared to the hierarchical trust model, the network trust model might better represent peer-to-peer business relationships whereby peers develop trust in each other through cross-certification instead of subordination.

The network trust model requires a globally accessible directory to distribute cross-certificates to PKI clients. Without a global directory, a PKI client cannot generally find the cross-certificates necessary to chain the certificate of a communicating peer to the CA that it trusts, causing the certificate verification process to fail. Note that a networked peer typically sends its certificate and the certificate of its own CA only when it communicates with another peer, which may have a different direct CA.

In the network of Figure 4-3, for example, a client of CA1 submits its certificate and the self-signed CA1 certificate when sending a secure S/MIME e-mail or an SSL message to a client of CA3. Clients of CA3, however, do not generally have local access to all the cross-certificates and must contact a global directory to build a proper certificate chain to their issuing CA.

The requirement for an on-line, globally accessible directory of cross-certificates introduces interoperability issues if a client cannot access the directory or if the directory does not contain an up-to-date list of cross-certificates. Furthermore, PKI clients may require application plug-ins because, in general, they do not know how to access a global directory and process cross-certificates. The distribution process of such plug-ins to all clients in a large network, especially an extranet, can become a deployment issue.

The Hybrid Trust Model

The hybrid trust model enables a rooted, hierarchical PKI to interoperate with a networked PKI. The network CAs must cross-certify with the hierarchy root CA; all the clients in the hierarchical PKI must add the network CAs to their list of trusted CAs. Baltimore Technologies [http://www.baltimore.com] is perhaps the most widely known PKI vendor that supports the hybrid trust model.

Certificate Chain Building

A certificate-based application must be able to chain a client's certificate to one of its trust points, typically a trusted, self-signed root CA, in order to verify the certificate and to develop trust in the claimed identity of the client. Some protocols, such as S/MIME and SSL, deliver complete certificate chains to their communicating parties, simplifying the certificate verification task. An application that does not receive a complete chain must build a chain of certificates between the end-entity certificate and a trust point.

Certificate chain building can be bottom-up or top-down. With **bottom-up chain building**, an application starts with an end-entity certificate and uses the information in the certificate to locate a parent certificate, iterating the process until reaching a trusted root. The bottom-up chain-building process is more efficient than the top-down approach and supports both off-line and on-line, dynamic certificate chain building. Rooted hierarchical PKIs typically use the bottom-up approach for discovering a certification path. Internet Explorer (IE), Internet Information Services (IIS), Outlook, and other CryptoAPI-based applications, for example, build bottom-up certificate chains.

With **top-down chain building**, an application starts with its directly trusted root CA and finds all the associated cross-certified CAs, repeating the process until finding the parent CA of the end-entity certificate. The application needs to access a global directory to retrieve the cross-certificates, unless it has local access to all of them. Networked PKIs generally use the top-down approach for certificate path discovery.

Windows 2000 builds a bottom-up certificate chain from an end-entity certificate up to a trusted CA root, as follows. (Refer to Chapter 3 for a description of the certificate extensions referenced.)

1. If the communicating application does not deliver a trusted certificate chain, Windows 2000 uses the authority key identifier (AKI) extension, if present, to find the parent certificate on the local computer. If the AKI extension contains both the key identifier field and the certificate issuer and serial number fields, Windows 2000 uses the issuer and serial number information first.
2. If it does not find a parent certificate on the local computer, based on the AKI information, Windows 2000 uses the location information in the authority information access (AIA) extension, if present, to retrieve the parent certificate from the network over HTTP, LDAP, or other transport protocols.
3. If steps 1 or 2 do not locate a parent certificate, Windows 2000 uses the issuer name information in a certificate to find a parent certificate on the local computer that has a matching subject name.

Revocation Status Checking

Windows 2000 supports certificate revocation lists (CRLs) as the primary mechanism for revocation status checking. Windows 2000 PKI uses locally cached CRLs; if it cannot find a suitable CRL or if the cached CRL has expired, Windows 2000 PKI dynamically obtains an up-to-date CRL over the network, using a variety of access protocols, such as LDAP, HTTP, and FILE. Windows 2000 uses the CRL distribution point extension for the on-line retrieval of CRLs; it does not support partitioned CRLs and fails revocation status checking.

Windows 2000 PKI does not support the Online Certificate Status Protocol (OCSP) [MYER99] for on-line revocation status checking, does not recognize authority revocation lists (ARLs), and expects a CRL to contain revocation information for both end-entities and certification authorities. An ARL is a special kind of CRL that contains revocation information for CAs only.

Cryptographic Algorithms and Key Lengths

Windows 2000 delivers two CSPs to implement the underlying capabilities of CryptoAPI: a base CSP and an enhanced CSP. Table 4-1 lists the various algorithms and key lengths supported by these two CSPs.

The Microsoft Enhanced CSP is available to all countries worldwide, except to U.S.-embargoed destinations. In addition to the export controls imposed by Microsoft, other countries may exercise separate jurisdictions over the import, export, or use of encryption products. Refer to [http://www.microsoft.com/exporting] for the most up-to-date information on the export control status of Microsoft products.

Table 4-1 Windows 2000–Supported Algorithms and Key Lengths

Algorithm	*Base CSP Key Length (bits)*	*Enhanced CSP Key Length (bits)*
MD2, MD5	√	√
SHA-1	√	√
MAC, HMAC	√	√
RSA Signing	Min: 384/Max: 16,384	Min: 384/Max: 16,384
RSA Key Exchange	Min: 384/Max: 1,024	Min: 384/Max: 16,384
DSA Signing	Min: 512/Max: 1,024	Min: 512/Max: 1,024
Diffie-Hellman Store and Forward	Min: 512/Max: 1,024	Min: 512/Max: 4,096
Diffie-Hellman Ephemeral	Min: 512/Max: 1,024	Min: 512/Max: 4,096
RC2, RC4, RC5	Min: 40/Max: 56	Min: 40/Max: 128
DES40	√	√
DES56	√	√
Triple DES EDE		√

Hardware Support

Windows 2000 uses CryptoAPI to provide an abstraction layer between applications and key stores. Software-based CSPs manage keys in software, whereas hardware-based CSPs use hardware tokens, such as smart cards, to store and to manage keys. Applications determine the required quality of protection for keys and choose appropriate CSPs.

Windows 2000 uses the PC/SC standard [http://www.pcscworkgroup.com] to interface with smart cards and smart card readers. This standard supports plug-and-play and power management features, which are essential to the Windows 2000 environment. Microsoft has no plans to add support for the PKCS #11 standard, which is an alternative protocol for communication with hardware devices.

Certificate Trust Lists

A **certificate trust list** (**CTL**) is a list of hashes of CA root certificates, digitally signed by a trusted administrator. A CTL has a validity period and contains restrictions that limit the scope of trust that a given CA has. CTLs provide a convenient, secure mechanism to distribute CA roots to PKI client applications. CTLs are useful in extranet scenarios in which an enterprise needs to establish a trust relationship with a partner without requiring the creation and management of cross-certificates.

Public Key Infrastructure Standards

Windows 2000 adheres to standards set forth by the International Telecommunication Union (ITU) and Internet Engineering Task Force (IETF) standards bodies to ensure maximum interoperability with third-party PKI vendors and service providers. These standards include X.509 V3 for public key certificates, CRL V2 for revocation information, TLS 1.0/SSL 3.0 for Internet security, and Kerberos V5 for network domain authentication. Table 4-2 lists the supported PKI standards.

Table 4-2 Windows 2000 PKI Standards

Standard	*Description*
X.509 V3	Defines standard format for public key certificate. Refer to [http://www.ietf.org/html.charters/pkix-charter.html] or Chapter 3 for details.
CRL V2	Defines standard format for certificate revocation lists. Refer to [http://www.ietf.org/html.charters/pkix-charter.html] or Chapter 3 for details.

(*continued*)

Table 4-2 Windows 2000 PKI Standards (*cont.*)

Standard	*Description*
PKIX	The Public Key Infrastructure Working Group of IETF (PKIX) [http://www.ietf.org/html.charters/pkix-charter.html] is in charge of defining an interoperable PKI for the Internet.
PKCS	De facto standards for public key message exchanges. Refer to [http://www.rsasecurity.com/rsalabs/pkcs] or Chapter 3 for details.
TLS/SSL	Provides a secure and authenticated channel between hosts on the Internet above the transport layer [http://www.ietf.org/html.charters/tls-charter.html].
IPsec	Defines transparent encryption of network traffic. Refer to [http://www.ietf.org/html.charters/ipsec-charter.html] or Part III for details.
Kerberos V5	Provides a symmetric key-based framework for authentication in large networks. See [http://www.ietf.org/html.charters/cat-charter.html] or Part I for details.
S/MIME	Provides a standard for secure e-mail in the Internet. [http://www.ietf.org/html.charters/smime-charter.html].
PC/SC	Provides a standard for integrating smart cards and smart card readers into a computing environment [http://www.pcscworkgroup.com].

Interoperability with Third-Party PKIs

Scalability and distributability are perhaps the biggest added-value propositions of a PKI-based authentication system. Using the Internet and PKIs as the backbone, millions of consumers around the globe authenticate hundreds of thousands of Web sites to conduct business-to-consumer processes. Similarly, business partners authenticate themselves to extranets for business-to-business functions. To PKI-enable the Internet, commercial third-party CA service providers supply outsourced trust services, third-party PKI product vendors deliver in-house solutions, and some operating system vendors provide built-in PKI support. To interconnect the maze of resulting trust paths, interoperability among disparate PKIs and PKI-enabled applications becomes critical.

We identify PKI-to-PKI, PKI-to-application, and application-to-application as three levels of interoperability. **PKI-to-PKI interoperability** allows a PKI system to transparently exchange authentication information with another PKI system. For example, a user in one PKI hierarchy can authenticate a user in another PKI hierarchy, regardless of the specific trust models. **PKI-to-application interoperability**

enables a PKI system to support applications designed for another PKI system, such as the Windows 2000 EFS application. **Application-to-application interoperability** addresses how two applications that conform to a standard can interoperate, regardless of their specific PKI environments. Interoperability issues between S/MIME e-mail applications and Web browsers to Web servers are examples.

Standards lay the groundwork for interoperability. Standards, however, are subject to various levels of conformance and interpretations, which results in interoperability issues. The RFC 2459 standard [HOUS99], for example, profiles the issuing distribution point (IDP) as a critical CRL extension that identifies the CRL distribution point for a particular CRL. The standard does not mandate that a vendor support this extension in order to be compliant.[6] Unsupported critical extensions, however, cause interoperability issues because an application that does not understand a critical extension must reject the certificate for integrity reasons; it cannot simply ignore it.

Although the RFC 2459 standard does provide a foundation for interoperability, we do not expect conforming PKI systems to have PKI-to-PKI interoperability. We do, however, expect to see a high level of interoperability across vertical applications—application-to-application interoperability—such as S/MIME and TLS/SSL, especially as vendors deploy their applications, gain experience on interoperability issues, and feed their findings back into standards. PKI-to-application interoperability is making headway as PKI vendors and application providers work out the integration issues.

PKI to PKI

A number of consumer scenarios require two PKIs to interoperate. A very large enterprise may be operating two or more PKIs supplied by different vendors across its organizational boundaries. The enterprise management now wishes to centralize its PKI operation and to provide PKI interoperability among its organizations. Company mergers may also necessitate PKI-to-PKI interoperability if two companies are running different PKI systems. Another scenario involves an enterprise that has and wants to upgrade an existing PKI to Windows 2000. The enterprise wishes to continue using the existing third-party PKI and also wants to leverage the built-in Windows 2000 PKI for a specialized purpose. In another, similar scenario, PKI-to-PKI interoperability issues surface when an enterprise that is using Windows 2000 PKI wishes to deploy a third-party PKI for a specialized use.

6. We must emphasize that the RFC 2459 standard attempts to solve a fairly general problem and needs to be broad in nature.

Numerous factors impact the interoperability of Windows 2000 PKI with a third-party PKI and ultimately PKI-to-application and application-to-application interoperability [MICR00C]. Some factors, such as the lack of a common trust model, have widespread impact and prevent proper authentication between two systems. Such factors as key import and export issues cause administrative overheads and some usability issues, but they do not cripple the entire system.

Trust Model

Windows 2000 supports the rooted hierarchical trust model. A third-party PKI designed around the network trust model faces fundamental PKI-to-PKI interoperability issues. Users in the network community cannot authenticate the users in the hierarchical community, because they do not have the necessary cross-certificates. The hierarchical community users fail to authenticate the network community users because they cannot construct a trust path to their trusted CAs. No quick solution exists to fix these problems; the third-party PKI needs to support the rooted hierarchical trust model in addition to its native network trust model.

Global Directory

The operation of a global directory is central to some third-party PKIs, especially those designed around the network trust model. The directory supplies CRLs and cross-certificates to the PKI clients, without which the PKI system cannot function. Windows 2000 PKI can make heavy use of Active Directory as the central data repository but does not depend on a global directory for proper functioning. Windows 2000 PKI can access CRLs and CA certificates from local caches or can retrieve them over the network by using appropriate certificate extension fields. Similarly, the Windows 2000 Certificate Services can work either in standalone mode, without Active Directory, or enterprise mode, requiring Active Directory.

PKIs that depend on a global directory do not necessary have interoperability issues with the Windows 2000 PKI. Active Directory is an integral part of Windows 2000 architecture and runs on every domain controller. The relevant issue is whether a third-party PKI can leverage Active Directory as the underling global repository or must install its own directory. A third-party PKI that requires its own global directory forces an enterprise to deploy and to manage a foreign directory in addition to its native Active Directory, causing administrative overheads and data synchronization issues.

Revocation Status Checking

When it checks a certificate chain for revocation information, the Windows 2000 PKI needs to access an up-to-date CRL for each certificate in the chain. If an appropriate CRL is not available on the local computer, Windows 2000 uses the uniform resource

identifier (URI) in the CRL distribution points (CDP) extension to dynamically retrieve the CRL. A third-party PKI that provides a value other than a URI in the CDP extension cannot interoperate with Windows 2000 PKI. Furthermore, vendors that do not support this extension may face interoperability issues unless client computers are synchronized with appropriate CRLs before Windows 2000 begins the revocation checking process.

The IETF RFC 2459 standard profiles the issuer distribution point (IDP) as a critical CRL extension, which identifies the CRL distribution point for a particular CRL. Windows 2000 PKI does not support critical CRL extensions and rejects CRLs that have the IDP extension set.[7] PKI vendors that use the IDP extension with their CRLs have interoperability issues with the Windows 2000 PKI.

Cryptographic Keys

The level of support a PKI provides for key migration across disparate PKIs and the assumptions a PKI makes about the number of keys an end-entity must possess for correct functions impact PKI-to-PKI interoperability and usability. The main areas of concern are (1) support for exporting and importing keys and (2) single versus dual key use.

It is desirable for users to have the same keys for an application that spans multiple PKIs. PKIs that do not support key import and export force their users to maintain multiple keys for an application, causing administrative overheads and potential security weaknesses. Windows 2000 supports the PKCS #12 standard for secure import and export of keys among PKIs, applications, and computer systems. Third-party PKI vendors that do not support PKCS #12 cannot migrate keys to or from the Windows 2000 PKI.

Windows 2000 PKI does not mandate whether an end-entity should have one certificate good for both signing and encryption or two separate certificates, one for signing and the other for encryption. The issues surrounding dual key pair versus single key pair invariably boil down to key escrow and key life-cycle management requirements. Windows 2000 supports both single key pair and dual key pair policies, allowing enterprises to determine their own policies on these issues.

Third-party PKI vendors that do not provide the same level of flexibility may run into PKI-to-application interoperability issues with Windows 2000 applications. Consider an S/MIME e-mail application, for example, that is designed or configured to work with single key pair certificates. When it receives an authenticated e-mail, this application can use the same certificate that signed the e-mail to send an encrypted e-mail to the sender. If a third-party PKI requires separate signing and encryption certificates, the S/MIME application can no loner use the signing certificate for sending

7. Note that implementations conforming to RFC 2459 [HOUS99] need not support the IDP extension.

an encrypted e-mail but must now find an appropriate encryption certificate, using another mechanism, such as a directory lookup.

Certificate Handling and Extensions

Similar to the cryptographic keys, two PKI systems with different certificate-handling capabilities may have interoperability issues. A PKI system that does not support Unicode name encoding, for example, may crash when it processes certificates with international names encoded in the Unicode format. Different design assumptions about certificate extensions and the presence of critical extensions in certificates cause major interoperability issues.

Windows 2000 PKI supports the Unicode format and can process certificates that have Unicode names in their subject, issuer, or subject alternative name fields. PKIs that do not support the Unicode name encoding or provide limited support face interoperability issues with Windows 2000 PKI.

Following is a list of certificate extensions and CRL extensions that have a widespread impact on interoperability. Note that although the subject alternative name and extended key usage extensions are more relevant to application-to-application interoperability, we present them here for structural reasons. Refer to Chapter 3 for an explanation of these extensions.

- The authority information access (AIA) extension indicates where to access information about the CA that has issued a certificate. This extension contains location information in the URL format, using various access methods, such LDAP, HTTP, and FILE. Multiple URLs in the AIA extension provide a level of fault tolerance.

  CryptoAPI may access the AIA extension during the dynamic chain-building process to retrieve parent CA certificates. The AIA extension is useful for Internet or extranet PKI deployments in which PKI clients may not have access to all the CA certificates and cannot access a central directory to retrieve them. The AIA extension provides an alternative approach whereby a PKI dynamically fetches parent certificates over a transport protocol. Third-party PKI vendors that do not populate the AIA extension limit the functionality of CryptoAPI and may cause a failure during the certificate chain-building process.

- Windows 2000 PKI uses the authority key identifier (AKI) extension during the chain-building process, as described in the section Certificate Chain Building. This extension provides key identifier, as well as serial and issuer information about the CA that has issued a certificate. Third-party PKI vendors that do not populate the AKI extension limit the functionality of Windows 2000 PKI.

- The CRL distribution points (CDP) extension indicates how to obtain revocation information for a certificate. Windows 2000 PKI looks up the value of this extension if it cannot find an up-to-date CRL on the local computer during revocation checking. Note that CRLs have validity periods and become obsolete after their end dates. This extension allows CryptoAPI to obtain a valid CRL over the network, using a variety of access protocols, such as LDAP, HTTP, and FILE.

  The CDP extension is extremely useful when a PKI needs to operate across organizational boundaries, when a central administration cannot synchronize client computers with the latest CRLs. PKI vendors that do not populate the CDP extension cause failures when Windows 2000 applications cannot find a valid CRL on the local computer during revocation checking. Furthermore, Windows 2000 PKI does not support X.500 distinguished names in the CDP extension and rejects certificates with such names in the CDP extension.

- The basic constraints extension indicates whether a certificate is an end-entity certificate or a CA certificate. Windows 2000 PKI checks this extension to ensure that an end-entity does not act as CA. Third-party PKIs that do not set this extension in their CA certificates cannot interoperate with the Window 2000 PKI.

- The subject alternative name contains additional name information about the subject name specified in a certificate, such as e-mail name, domain name system (DNS) name, and user principal name (UPN). Some applications, such as S/MIME and IPsec, may not function correctly without this extension. Windows 2000 populates the subject alternative name field when issuing S/MIME and IPsec certificates. PKI vendors that do not populate this extension cause application-to-application interoperability issues.

- The extended key usage (EKU) extension provides further information for the valid use of a certificate and further constraints on the key usage extension. This extension, if marked critical, specifies that a CA has issued a certificate for a particular purpose and that a relying party should not accept it for other purposes. Applications should check the EKU extension and reject certificates that are not valid for a particular function. An S/MIME application, for example, should reject certificates that are issued for File System Encryption (FES) only.

  Certificates that do not contain the EKU extension do not typically cause interoperability problems unless a relying application is especially careful and strictly enforces the right usage of certificates, perhaps to meet special integrity requirements. A certificate that does not contain the EKU extension is valid for all usage as specified by the key usage extension. Windows 2000 supports the EKU extension and uses it to select an appropriate certificate when a user has a number of certificates.

PKI to Application

PKI-to-application interoperability deals with the ability of a PKI system to support applications designed to work with another PKI system. The customer scenarios discussed in the previous section may also require PKI-to-application interoperability. A large enterprise that is running various PKI systems across its organizational boundaries may want to standardize on one PKI system that supports all its certificate-using applications. Similarly, a merger between two companies may require one PKI system to provide public key capabilities to both companies for all applications. Companies that are upgrading to Windows 2000 face similar issues if they are already using another PKI system.

PKI-to-application interoperability with Windows 2000 PKI involves issues about the ability of a third-party PKI to support certificate-using applications in Windows 2000 and the ability of the Windows 2000 PKI to support third-party certificate-using applications. This section focuses on the more practical case of an enterprise that wants to use a third-party PKI in a Windows 2000 environment exclusively; it is less likely that an enterprise wants the Windows 2000 PKI to run specialized, third-party PKI applications.

Windows 2000 applications have various levels of interoperability with a third-party PKI. Some applications, such as secure Web and secure e-mail, score at the top of the interoperability spectrum, whereas some other applications, such as EFS and smart card log-on, face a number of interoperability issues.

Secure Web

Third-party PKIs have been issuing SSL server certificates for Microsoft Internet Information Services (IIS) for a number of years now and can continue to do so; there should be no interoperability issues with the Windows 2000 IIS. Windows 2000 embeds the root certificates of more than 100 commercial third-party CAs, representing more than 20 countries. These root certificates are enabled for server authentication by default. Among all the certificate-based applications, Web servers have probably the best level of interoperability with third-party PKIs and browsers, across various deployment scenarios. Furthermore, third-party PKIs can issue client certificates for Web access control without any interoperability issues.

Possible interoperability issues center on certificate revocation checking and a rather strange problem with TLS/SSL client-side authentication. As already mentioned, Windows 2000 PKI does not support critical extensions on CRLs, specifically the issuer distribution point extension. IIS and Internet Explorer fail revocation status checking if a CRL contains a critical extension and reject a certificate that might otherwise be valid.

Windows 2000 does not enable client authentication by default for all the preconfigured commercial CAs. Some TLS/SSL implementations have interoperability

issues when the list of CAs trusted for client authentication is long and fragments over multiple network packets. These implementations cannot handle the fragmentation and fail the TLS/SSL handshake protocol. An enterprise that uses client certificates for Web access control needs to ensure that the issuing CA is enabled for client authentication.

Secure E-Mail

Similar to the secure Web servers and browsers, secure e-mail applications based on S/MIME are highly interoperable across various PKIs and deployment configurations. Windows 2000 enables the preconfigured list of trusted commercial CAs for secure e-mail by default. Interoperability issues may arise, however, if a PKI does not populate the alternative subject name extension, does not support the Unicode name format, cannot handle large RSA keys, or mandates the presence of CA certificates in a global directory. Furthermore, critical extensions on CRLs introduce interoperability issues with Windows 2000 Outlook and Outlook Express.

Encrypting File System

A third-party PKI can issue certificates for Windows 2000 EFS application under the following two conditions. First, it must populate the EKU extension to indicate that the issued certificate is valid for EFS. Second, it must use the Microsoft Base Cryptographic Service Provider (CSP) to manage the keying material. A third-party CA can either directly generate the key pair on the Microsoft Base CSP or use PKCS #12 to import the key pair to the Microsoft Base CSP.

Windows 2000 PKI seamlessly integrates EFS within the operating system. EFS automatically obtains an EFS certificate for a user from a Windows 2000 CA, if present, or issues its own self-signed certificate. EFS will not automatically interface with a third-party CA to issue EFS certificates for users.

Certificate Mapping

Windows 2000 certificate mapping allows an enterprise to use the Internet for expanding its reach to its business partners, suppliers, customers, and the general Internet public. The enterprise defines a mapping between a certificate and a Windows 2000 account by creating a mapping entry in either the Internet Information Services or Active Directory. Distributed partners use their certificates to access extranet resources, subject to Windows 2000 access-control verifications. Certificate mapping works with certificates issued by any third-party PKI.

Machine Autoenrollment

Computers running Windows 2000 can automatically enroll for certificates against a Windows 2000 Certificate Services enterprise CA. The machine autoenrollment

feature does not work with third-party PKIs. A third-party CA, however, can be the parent CA of a Windows 2000 CA that issues machine certificates, as explained in Chapters 5 and 9.

Smart Card Log-On

Windows 2000 leverages the public key extensions of Kerberos to enable smart card–based interactive log-on to a domain network, as described in Chapter 2. A smart card used for network log-on must contain a certificate issued by a Windows 2000 Certificate Services enterprise CA. Windows 2000 rejects certificates that are issued by any other CA. The requirement to have a Windows 2000 CA as the certification authority for smart card certificates prevents an attacker from breaking into another company's network by using a foreign PKI. Although a third-party CA cannot issue log-on smart card certificates, it can be the parent CA of a Windows 2000 CA that issues such certificates.

Application to Application

A PKI system provides public key capabilities to applications. PKI-to-PKI and PKI-to-application interoperability issues ultimately determine whether two applications, possibly running across different organization boundaries with different underlying PKI systems, can communicate according to the intended design principles and integrity parameters. The capabilities of a PKI for supporting various cryptographic algorithms and cryptographic strengths influence application-to-application interoperability. The correct functioning of an application may also hinge on certain certificate extensions, such as the subject alternative name extension, as discussed previously.

The cryptographic algorithms and the key lengths supported by a PKI can directly impact application-to-application interoperability. Windows 2000 uses CryptoAPI in conjunction with the underlying Microsoft Base and Microsoft Enhanced CSPs to provide cryptographic capabilities to applications. Similarly, other third-party PKI vendors provide an engine to provide such capacities to their applications. Applications will fail to interoperate if they cannot negotiate a common encryption algorithm and key length, using such protocols as TLS/SSL. Similarly, store-and-forward applications, such as e-mail, may fail if a sender and a receiver use PKIs with different capabilities. An e-mail application that has low-strength cryptographic capabilities, for example, cannot decrypt an e-mail encrypted with triple DES.[8]

8. It should be noted that Microsoft Mail applications, both the original Exchange Mail client and Outlook, are able to decrypt all strengths of encryption; the only issue is at what level they are able to encrypt messages. This capability removes this issue as a problem for Microsoft Mail applications. We thank Christopher Lowde for this clarification.

Summary

Windows 2000 leverages public key technology to address the security needs of enterprises and e-commerce applications. These security needs include authenticating distributed business partners, using smart cards for strong network authentication, distributing authenticated code, and ensuring laptop and desktop file system security, secure e-mail, and network-level secure communications.

The public key security architecture of Windows 2000 supports e-commerce applications that require distributed, scalable authentication. The components of this architecture seamlessly integrate with the operating system, instead of being an adjunct to it, and effectively interface with external trust infrastructures. Microsoft CryptoAPI, the cornerstone of this architecture, provides the machinery to build authentication, integrity, and privacy security services into applications. At the very top are a myriad of PKI-enabled applications that layer on top of CryptoAPI. These applications leverage the Window 2000 PKI to establish the quality of protection they need before they process a transaction or open a communication channel.

Standards lay the groundwork for interoperability among disparate PKI systems. Standards, however, are subject to various levels of conformance and interpretations, which result in interoperability issues. PKI-to-PKI, PKI-to-application, and application-to-application are three main areas of interoperability concerns.

References

[DES83] *American National Standard for Information Systems-Data Link Encryption,* ANSI X3.106, American National Standards Institute, 1983.

[DIER99] Dierks, T., and C. Allen, *The TLS Protocol Version 1.0,* Internet Request for Comments (RFC) 2246, January 1999. (http://www.ietf.org/rfc/rfc2246.txt)

[DSS94] *Digital Signature Standard (DSS),* FIPS PUB 186, National Institute of Standards and Technology, U.S. Department of Commerce, May 1994.

[FEGH98] Feghhi, J., J. Feghhi, and P. Williams, *Digital Certificates: Applied Internet Security,* (Reading, MA: Addison-Wesley, 1998). (http://cseng.awl.com/bookdetail.qry?ISBN=0-201-30980-7)

[FRIE96] Frier, A., P. Karlton, and P. Kocher, *The SSL 3.0 Protocol,* Netscape Communications, Nov. 18, 1996.

[HOUS99] Housley, R., W. Ford, W. Polk, and D. Solo, *Internet X.509 Public Key Infrastructure Certificate and CRL Profile,* Internet Request for Comments (RFC) 2459, January 1999. (http://www.ietf.org/rfc/rfc2459.txt)

[KRAW97] Krawczyk, H., M. Bellare, and R. Canetti, *HMAC: Keyed-Hashing for Message Authentication,* Internet Request for Comments (RFC) 2104, February 1997. (http://www.ietf.org/rfc/rfc2104.txt)

[MICR00A] Microsoft Corporation, "Microsoft Windows 2000 Public Key Infrastructure," white paper, 1999. (http://www.microsoft.com/windows2000/library/planning/security/pki.asp)

[MICR00B] Microsoft Corporation, "An Introduction to the Windows 2000 Public-Key Infrastructure," white paper, 1999. (http://www.microsoft.com/windows2000/library/howitworks/security/pkiintro.asp)

[MICR00C] Microsoft Corporation, "Public Key Interoperability," white paper, 1999. (http://www.microsoft.com/windows2000/library/howitworks/security/w2kpkint.asp)

[MICR00D] Microsoft Corporation, "Encrypting File System for Windows 2000," white paper, 1999. (http://www.microsoft.com/windows2000/library/howitworks/security/encrypt.asp)

[MYER99] Myers, M., R. Ankney, A. Malpani, S. Galperin, and C. Adams, *X.509 Internet Public Key Infrastructure Online Certificate Status Protocol—OCSP,* Internet Request for Comments (RFC) 2560, June 1999. (http://www.ietf.org/rfc/rfc2560.txt)

[RIVE78] Rivest, R., A. Shamir, and L. M. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," *Communications of the ACM,* 21(2), February 1978, pp. 120–126.

[RIVE92] Rivest, R., *The MD5 Message-Digest Algorithm,* Internet Request for Comments (RFC) 1321, April 1992. (http://www.ietf.org/rfc/rfc1321.txt)

[SHA94] *Secure Hash Standard,* NIST FIPS PUB 180-1, National Institute of Standards and Technology, U.S. Department of Commerce, Work in Progress, May 31, 1994.