RSA Data Security®
A Security Dynamics Company

Understanding Public Key Infrastructure (PKI)

An RSA Data Security White Paper

RSA Data Security, Inc.
2955 Campus Drive, Suite 400
San Mateo, CA 94403-2507
Phone: 650-295-7600
Fax: 650-295-7700
Web: www.rsa.com

THE KEY MANAGEMENT PROBLEM

The Need for Public Key Cryptography

The way the world does business is changing, and corporate security must change accordingly.

For instance, e-mail now carries not only memos and notes, but also contracts and sensitive financial information. The Web is being used not only for publishing corporate brochures but for software distribution and e-commerce. Virtual private networks (VPNs) are extending corporate networks onto the Internet. Extranets turn the Internet into a selectively shared VPN.

Secure e-mail, Web access, e-commerce, VPNs and extranets require strong security, which provides confidentiality, authentication, access control, data integrity, and accountability. Certificates and public key cryptography are emerging as the preferred enablers of strong security. Many large organizations will deploy public key cryptography and certificates throughout the company in the next few years.

Public key cryptography requires a *public key infrastructure* (PKI), essential services for managing digital certificates and encryption keys for people, programs and systems.

What is Public Key Cryptography?

Cryptography uses mathematical algorithms and processes to convert intelligible *plaintext* into unintelligible *ciphertext*, and vice versa. Applications of cryptography include:

- Data encryption for confidentiality
- Digital signatures to provide non-repudiation (accountability) and verify data integrity
- Certificates for authenticating people, applications and services, and for access control (authorization)

There are two main kinds of cryptography: *shared secret* and *public key*.

In shared secret cryptography, sender and receiver use the *same* key for both encryption and decryption. Thus, many clients need to have the same key. Since encryption is presumably not available prior to key distribution, network-based key distribution is not a secure option. Other options, such as a secure courier, are expensive and slow.

Public key cryptography, in contrast, uses pairs of keys: a *public* key that is widely available, and a different *private* key known only to the person, application or service that owns the keys. The public key can be transmitted unencrypted over insecure lines, since it is not a secret, while the private key must be kept secret. Thus, key distribution is greatly simplified using public key cryptography.

The sender's private key may be used to produce a *digital signature*, an encrypted block of data which, when decrypted by the recipient, verifies the sender's identity (non-repudiation) as well as the integrity of the data. Some organizations issue a special signing key pair. Unlike encryption keys, signing keys are not archived, reducing exposure to unauthorized access to the private key.

Public key cryptography can also be used for secure distribution of shared secret keys across insecure networks. In each of these applications, access to the public key never gives a perpetrator access to protected data. For more information, see "Frequently Asked Questions About Today's Cryptography" on RSA's Web site (http://www.rsa.com/rsalabs/faq).

Certificates Validate Public Keys

Strong public keys as "read only" objects in secure directory services provides some protection against switched or forged keys. However, a perpetrator could pretend to be the secure directory service and give a user forged public key, allowing the perpetrator, and not the intended recipient, to decrypt messages encrypted with the key. Thus, a means of validating public keys is required. Certificates provide key validation.

A certificate is a digital document (*i.e.* a formatted file) that binds a public key to a person, application, or service. A trusted Certificate Authority (CA) creates the certificate and digitally signs it using the CA's private key. Because of its role in creating certificates, the CA is the central component of the PKI. Using the CA's public key, applications verify the issuing CA's digital signature, and hence, the integrity of the contents of the certificate (most importantly, the public key and the identity of the person, application, or server).

Many applications: many keys and certificates

Initially, many organizations manage keys and certificates manually, and on an application-by-application basis. This works acceptably when organizations deploy applications to moderate numbers of users in a small number of applications. However, as public key usage grows, management tasks become more challenging. Most organizations evolve to an automated, consolidated approach, based on a PKI.

PKI COMPONENTS AND FUNCTIONS

There are three core functional components to a PKI:

- The *Certificate Authority (CA)*, an entity which issues certificates. One or more in-house servers, or a trusted third party such as VeriSign or GTE, can provide the CA function.
- The *repository* for keys, certificates and Certificate Revocation Lists (CRLs) is usually based on an Lightweight Directory Access Protocol (LDAP)-enabled directory service.
- A *management function*, typically implemented via a management console.

If the PKI provides automated key recovery, there may also be a *key recovery service*. Key recovery is an advanced function required to recover data or messages when a key is lost.

In addition, there may be a separate *Registration Authority (RA)*, an entity dedicated to user registration and accepting

requests for certificates. User registration is the process of collecting user information and verifying user identity, which is then used to register a user according to a policy. This is distinct from the process of creating, signing, and issuing a certificate. The Human Resources department may manage the RA function, for instance, while Information Technology manages the CA. A separate RA also makes it harder for any

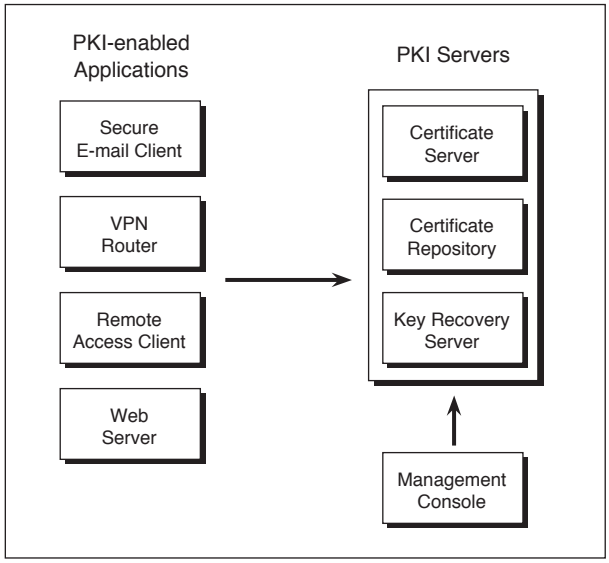

Figure 1. The main server components of a PKI are the certificate server, the repository, the key recovery server, and usually accompanied by a management console.

single department to subvert the security system. Organizations can choose to have registration handled by a separate RA, or included as a function of the CA.

Note that the CA, RA and so on are functional components that may be implemented in various ways in hardware and software. In particular, the CA can be implemented on one or more servers, as can the RA. Systems performing CA and RA functions are often referred to as Certificate Servers and Registration Servers respectively.

PKI Functions

The most common PKI functions are issuing certificates, revoking certificates, creating and publishing CRLs, storing and retrieving certificates and CRLs, and key lifecycle management. Enhanced or emerging functions include time-stamping and policy-based certificate validation. (PKI functions are summarized below in Table 1.)

Issuing certificates

The CA signs the certificate, thereby authenticating the identity of the requestor, in the same way that a notary public vouches for the signature and identity of an individual. In addition, the CA "stamps" the certificate with an expiration date. The CA may return the certificate to the requesting system and/or post it in a repository.

Revoking certificates

A certificate may become invalid before the normal expiration of its validity period. For instance, an employee may quit or change names, or a private key may be compromised. Under such circumstances, the CA revokes the certificate by including the certificate's serial number on the next scheduled CRL.

Storing and retrieving certificates and CRLs

The most common means of storing and retrieving certificates and CRLs is via a directory service, with access via LDAP. Other options include X.500 compatible directories, HTTP, FTP, and e-mail.

Providing trust

Each public key user must have at least one public key from a CA that the user trusts implicitly. Organizations can establish and maintain trust within a single security management domain through a thorough audit of the CA's policies and procedures, repeated at regular intervals.

However, organizations need to evaluate (and accept or reject) certificates from CAs not under their direct control, such as CAs of other business units or partners. This can be

Function	Description	Implementation
Registering users	Collect user information and verify user identity	Function of the CA, or a separate RA
Issuing certificates	Create certificates in response to a user or administrator request	Function of the CA
Revoking certificates	Create and publish Certificate Revocation Lists (CRLs)	Administrative software associated with the CA
Storing and retrieving certificates and Certificate Revocation Lists (CRLs)	Make certificates and CRLs conveniently available to authorized users	The repository for certificates and CRLs is usually a secure, replicated directory service accessible via LDAP
Policy-based certificate path validation	Impose policy-based constraints on the certificate chain, and validate if all constraints are met	Function of the CA
Time-stamping	Put a time-stamp on each certificate	Function of the CA or a dedicated Time Server (TS)
Key lifecycle management	Update, archive and restore keys	Automated in software or perfomed manually

Table 1. Public Key Infrastructure (PKI) Functions

accomplished through hierarchical *certification path processing* or direct *cross-certification*.

Certification Path Processing

The best known hierarchical certification path processing architectures are those maintained by PKI service organizations such as VeriSign. Typically, in such a hierarchy:

1) There is a single *root* at the top.
2) The root certifies *public primary certification authorities (PCAs)*, which issue, suspend, and revoke certificates for all CAs within the hierarchy.
3) PCAs certify *CAs*. PCAs may also cross-certify with PCA-like entities in other vendors' PKIs.
4) CAs authorize *subordinate CAs*, which belong to the PKI service company or the customer.
5) At the bottom of the hierarchy can be *local registration authorities (LRAs)* that evaluate certificate applications on behalf of the root, PCA, or CA that issues the certificates.

If a user does not already trust the CA that signed a certificate, the user searches upward through the hierarchy for a trusted CA that has certified the public key of the CA in question.

Cross-Certification

One CA can issue another CA a certificate that allows the other CA to issue certificates which will be recognized by the first CA. Cross-certification works directly, without a third party.

Hierarchical and Cross-Certification Can Be Combined

It can make sense to implement both hierarchical and cross-certification in a single security domain, for different purposes or at different times. For instance, a hierarchical system based on a trusted third party may be necessary when setting up or expanding a PKI, but the bulk of day-to-day interoperability may be accomplished via cross-certification.

Time-Stamping

In addition to the content and authenticity of a transaction, the exact time of the transaction can be important. For instance, a transaction may have to be submitted by a certain time to be valid. The solution to having trusted transactions is to combine digital signatures with time-stamps. This can be accomplished by augmenting a PKI with a time-stamping service.

Policy-based certificate path validation

Evaluating a certificate may require searching through a long chain of CAs. Ideally, every CA in the path would support any policy requirements associated with the validation. Such *policy-based certificate path validation* is not yet a standard part of CA functionality. However, some vendors are beginning to support it, and work is proceeding to define standards in this area.

Policy mapping matches one CA's policy to another, allowing CAs to interoperate if they support similar but not necessarily identical policies. It may be desirable to prohibit policy mapping, so that the path will be validated only if the CAs have identical policies.

Key lifecycle management

The PKI performs some functions, such as issuing a certificate and listing a certificate on a CRL, in response to a current request. In contrast, key lifecycle functions, such as updating, backing up, and archiving keys, are performed routinely.

Each user is likely to have a number of keys that require lifecycle management. For instance, users typically have at least one key pair for each secure application (*e.g.* e-mail, desktop file encryption, VPN). Some applications use several key pairs for different purposes, such as digital signatures, bulk encryption, and authentication.

Updating Keys

New keys are usually issued at regular intervals, such as every year or two, to reduce the exposure from keys that have been unknowingly compromised.

Backing Up Keys

Users frequently forget the passwords that protect their private keys; or they may "lose" the keys, for example, through a disk crash or virus attack. The company should be able to restore the keys to the user.

Archiving Keys

When employees leave the company, the network manager invalidates their encryption keys for future use, while retaining the keys in order to access previously encrypted files and email messages. Keys used for digital signatures may be retained for as long as the signed documents, so signatures can be verified. Note that many organizations archive encryption keys but not signing keys, since the availability of archived signing keys could invite abuse via identity impersonation.

Automated Key Lifecycle Management: A Critical PKI Function

The effort required to manage keys manually can limit the scalability of the PKI. Automated key management is a critical function for a large PKI.

HOW APPLICATIONS WORK WITH A PKI

The PKI manages the keys and digital certificates used to implement cryptography within applications such as e-mail and messaging, Web browsers and Web servers, electronic data interchange (EDI); in applications that establish secure network transactions or communications sessions over the Web or in VPNs using protocols such as S/MIME, SSL and IPSEC; and in functions such as digitally signed documents or code. In addition, applications developed in-house can be PKI-enabled.

E-Mail and Messaging

Secure e-mail and messaging use key pairs for encryption of messages and files, and for digital signatures. For instance, e-mail programs like Microsoft Exchange and IBM's Notes Mail are increasingly using encryption to carry sensitive information. The same is true of messaging-based groupware programs, such as Novell's GroupWise. EDI systems support financial transactions that require authentication, privacy and data integrity.

The most common secure e-mail/messaging protocol is Secure Multipurpose Internet Mail Extensions (S/MIME), which extends the Multipurpose Internet Mail Extensions (MIME) standard.

Web Access

Browsers and Web servers use encryption for authentication and confidentiality, and for applications like online banking and online shopping. Typically, using Secure Sockets Layer (SSL), servers authenticate themselves to clients. SSL also encrypts traffic. Client authentication is also an option. SSL is not limited to Hypertext Transfer Protocol (HTTP) but also supports protocols such as File Transfer Protocol (FTP) and Telnet.

VPN

Encryption and authentication are the main technologies used to convert standard Internet links into Virtual Private Networks (VPNs), used either for site-to-site privacy (router-to-router) or for secure remote access (client-to-server). These functions are implemented in the context of a tunneling protocol that wraps (or "encapsulates") one protocol in another protocol. For instance, the encapsulated protocol may be Point-to-Point Protocol (PPP), while the encapsulating protocol may be Internet Protocol (IP). The emerging standard for site-to-site tunneling is the IP Security (IPSEC) protocol from the IETF.

Digitally Signed Code and Files

Increasing reliance on downloaded programs and files has raised security concerns, particularly in the area of virus control. Technologies like Microsoft's Authenticode use RSA digital signatures to verify the source and the integrity of the content. A PKI is utilized in order to scale this approach to the huge numbers of users and programs requiring such services.

PKI-RELATED STANDARDS

Standards in the PKI arena fall into two groups: those that specifically define the PKI, and user-level standards that rely on the PKI, but don't define it.

Figure 2 shows the relationship between applications and infrastructure, and their associated standards.

PKI Standards

PKI standards permit multiple PKIs to interoperate, and multiple applications to interface with a single, consolidated PKI.

In particular, standards are necessary for:

- Enrollment procedures
- Certificate formats
- CRL formats
- Formats for certificate enrollment messages (client requests certificate, server issues certificate)
- Digital signature formats
- Challenge/response protocols

The primary focus of interoperable PKI standards is the PKI working group of the Internet Engineering Task Force (IETF), known as the PKIX group (for "PKI for X.509 certificates").

PKIX Overview

The four main components in the PKIX model are: the user (or "end entity"), CA, RA and repository.

Component	Part of PKI?	Description
User (end entity)	No	User of PKI certificates and/or end user system that is the subject of a certificate
Certification Authority (CA)	Yes	Issues, stores and revokes certificates
Registration Authority (RA)	Yes	An optional system to which a CA delegates certain management functions such as registering users
Repository	Yes	A system or collection of distributed systems that store and allow end entities to access certificates and CRLs.

Table 2. The main components of the PKIX model

PKIX Component Standards

The PKIX specifications are based on two other standards: X.509 from the International Telecommunication Union (ITU) and the Public Key Cryptography Standards (PKCS) from RSA Data Security. X.509 was intended to specify authentication services for X.500 directory services. In fact, the certificate syntax of X.509 has been widely adopted outside X.500 environments. However, X.509 was not intended to define a complete, interoperable PKI. To supplement X.509, vendors, users and standards committees have turned primarily to de facto PKI standards defined in PKCS.

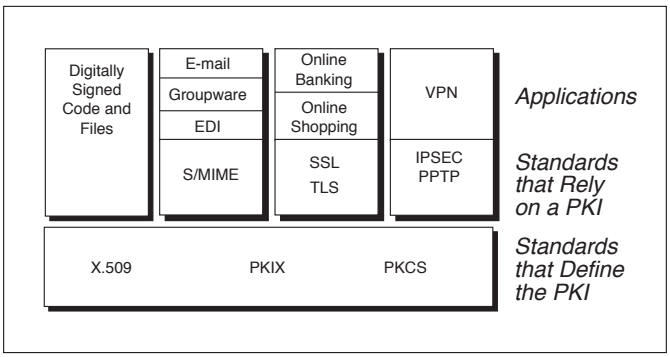

Figure 2. PKI standards define the PKI. Security standards for applications may require, assume or allow the use of a PKI..

X.509

The foundational and most universally supported PKI standard is the ITU's X.509. The primary purpose of X.509 is to define a standard digital certificate format.

PKCS

PKCS is a series of standards covering PKI in areas of certificate enrollment and renewal, and CRL distribution. For PKI interoperability, the three most important PKCS standards are PKCS #7, "Cryptographic Message Syntax Standard," PKCS #10, "Certificate Request Syntax Standard," and PKCS #12, "Personal Information Exchange Syntax Standard".

Standards That Rely on a PKI

Most major security standards are designed to work with a PKI. For instance, Secure Sockets Layer (SSL), Transport Layer Security (TLS), Secure Multipurpose Internet Mail Extensions (S/MIME), Secure Electronic Transactions (SET) and IP Security (IPSEC), all assume, require or allow the use of a PKI.

S/MIME

S/MIME is the IETF standard for secure messaging. S/MIME assumes a PKI for digitally signing messages and to support encryption of messages and attachments, without requiring prior shared secrets. Because e-mail is the most mature of the popular Internet applications, the S/MIME committee has led the way in implementing and extending PKI standards, taking advantage of the PKIX standards when possible, and filling in where additional standards were necessary. The most important standards developed by the S/MIME committee are Cryptographic Message Syntax, Message Specification, Certificate Handling, and Certificate Request Syntax.

SSL and TLS

SSL and the emerging IETF standard, TLS, which is based on SSL, are the most important standards for providing secure access to Web servers. SSL and TLS are also being used for general client/server security in a variety of non-Web applications. Both rely on a PKI for certificate issuance for clients and servers.

Secure Electronic Transactions (SET)

SET facilitates secure electronic bank card payments. SET uses keys for authentication, confidentiality and data integrity. PKI is a critical underpinning for authentication of the parties involved in a payment transaction.

IPSEC

The IETF Internet Protocol Security Protocol (IPSEC) standard defines protocols for IP encryption, and is one of the primary protocols used for deploying VPNs. IPSEC requires keys for encryption and authentication. Complete PKI standards for IPSEC are emerging, and a PKI is the most scaleable way of managing IPSEC keys. Use of IPSEC is still fairly limited. However, the need for PKI will grow with IPSEC deployment.

ISSUES IN PKI DEPLOYMENT

Five important questions need to be considered in deploying a PKI:

- What is the organization's PKI strategy?
- How will interoperability be achieved?
- Are applications PKI-ready?
- How many clients will be involved in the initial deployment?
- What are the technical staff requirements for deployment planning and for deployment?

What is the organization's PKI Strategy?

PKI strategy typically focuses either on enabling a specific application or on consolidating PKI functions for multiple applications.

Enabling a specific application

Many organizations do not currently have any applications that rely on a PKI. In that case, the most pragmatic approach, with the least investment and the quickest payback, is to focus on deploying a PKI for a single application. Experience gained with that application will make planning for the next application, as well as for a consolidated PKI, more fruitful.

Even when deploying a PKI for a single application, organizations typically want to consider how they will achieve interoperability for multiple applications, or PKI consolidation. (See the next question, "How will interoperability be achieved?")

Consolidating PKI functions for multiple applications

Organizations with several PKI-enabled applications can consolidate the PKI infrastructure, for management efficiency and cost savings. Organizations may also decide to implement the infrastructure first and then the applications if their strategy is to deploy multiple PKI-enabled applications as quickly as possible. However, implementing a complex system of this type without first solving problems associated with each application requires extra planning, piloting, and often outside assistance.

How will interoperability be achieved?

There are two basic approaches to PKI interoperability:

1) Focus on a particular vendor's product
2) Focus on standards

In the past, PKI standards were so immature that organizations often had to focus on PKI products from a single vendor. However, a single-vendor strategy is unlikely to prove viable on an enterprise level. Different companies or business units will want to interoperate even though they have implemented PKIs based on different vendors' products. PKI standards, with the PKIX specifications as a foundation,

have evolved to the point where companies can plan to use them as a framework for enterprise-level PKI interoperability. In a maturing, expanding PKI market, vendor-independent standards will increasingly be the method of choice for achieving interoperability and consolidation.

Are applications PKI-ready?

Very few applications are PKI-ready today. In most cases, organizations have two options:

1) *Encourage software vendors to PKI-enable their applications*. For user organizations, this is easy and requires no up-front investment. It typically also yields PKI capabilities that are smoothly integrated with applications. However, the timetable for delivery of PKI features is out of the user organization's control, and PKI features may not ideally fit the user organization's requirements.

2) *Use in-house programming staff or contract programmers to PKI-enable applications*. This should yield results that precisely suit the organization's needs. It also allows very smooth integration of PKI functions with in-house applications. In the case of commercial off-the-shelf (COTS) software, the organization is dependent on the APIs exposed by the software developer. These may vary in their suitability to PKI integration, and to the user organization's particular needs. Organizations planning to customize applications may wish to evaluate RSA's enabling tools and PKI servers, which may reduce the time and cost of the project.

How many clients will be deployed initially?

Vendors may imply that deployment of thousands or even tens of thousands of PKI clients is a reasonable first step. In reality, most organizations pilot with no more than a few hundred - and often fewer than a hundred - clients. There is usually little to be gained from rushing to deploy thousands of clients without making sure that the first few hundred are operating correctly.

What are the technical staff requirements?

According to the Aberdeen Group, less than half the cost of deploying a PKI is attributable to acquiring and installing hardware and software. In fact, technical personnel required for planning and deployment typically represent the largest cost. Companies address technical staff requirements either by staffing up internally or by outsourcing to professional services organizations. Generally, it makes most sense to staff up internally only for ongoing needs. One-time tasks may be better outsourced to an experienced partner that can perform them more efficiently, without the expense of hiring and laying off personnel.

CONCLUSION

Public key cryptography and certificates are emerging as the preferred enablers of strong security for a number of applications, including e-mail, Web access, VPNs and digitally signed code. A PKI manages keys and certificates for people, programs and systems. PKI standards, such as the PKIX specifications, allow multiple PKIs to interoperate, or multiple applications to use a single PKI. This makes PKI consolidation possible, facilitating a manageable, scaleable PKI. To successfully deploy a PKI, organizations must develop a sound strategy, plan for interoperability, determine how applications will interface with the PKI, size the initial project correctly, and plan for technical staff requirements.

RSA SECURITY PRODUCTS

RSA is a leading provider of software security components and technology. Over 400 innovative software vendors and enterprises utilize RSA software in their applications.

RSA's Integrated Security System

The RSA BSAFE Security Component Line is a set of complementary security products that provide complete security coverage for your applications. The BSAFE product line provides C and Java developers with implementations of critical security protocols (SSL-C, SSL-J, and S/MIME-C), core cryptographic services (Crypto-C and Crypto-J), and certificate processing services (Cert-C and Cert-J). RSA's BSAFE Security Components are specifically designed to be PKI-ready, speeding development of applications integrated with a public key infrastructure.

The Keon Product Family

Keon is a suite of products that work together to protect your applications. *Keon Certificate Server* provides a powerful and flexible system for issuing and managing digital certificates. In addition to *Keon Certificate Server*, the suite includes *Keon Security Server* for centralized security administration, user management and access control; *Keon Desktop* for desktop file encryption, managing single sign-on and user credentials at the desktop, and delivering services for securing email, Web browsers and access to applications; *Keon Agents* which are installed on protected application servers, and create the authenticated, secure connection from the desktop to the application server; and *Keon Agent SDK* which gives developers the ability to build Agents for in-house applications.

RSA and Security Dynamics also offer a range of enhanced PKI services based on the core PKI. For more information, contact RSA directly, or see RSA's Web site (http://www.rsa.com).