Public Key Infrastructure Overview

By Joel Weise - SunPS℠ Global Security Practice

Sun BluePrints™ OnLine - August 2001

http://www.sun.com/blueprints

Sun Microsystems, Inc.
901 San Antonio Road
Palo Alto, CA 94303 USA
650 960-1300 fax 650 969-9131

Part No.: 816-1279-10
Revision 01, August 2001

Public Key Infrastructure Overview

Complex business systems, e-commerce and automated business transactions require robust and rigorous security measures. Companies using the Internet environment as a platform to conduct business have a better probability of success if they accommodate the needs of security-conscious clientele. Today's Internet clientele demand stringent security protocols to protect their interests, privacy, communication, value exchange, and information assets. This article demonstrates how public key cryptography supports these risk management requirements and solves e-commerce security problems in heterogeneous network environments.

Public key cryptography supports security mechanisms such as confidentiality, integrity, authentication, and non-repudiation. However, to successfully implement these security mechanisms, you must carefully plan an infrastructure to manage them. A public key infrastructure (PKI) is a foundation on which other applications, system, and network security components are built. A PKI is an essential component of an overall security strategy that must work in concert with other security mechanisms, business practices, and risk management efforts.

PKI is a broad subject matter and is constantly evolving to meet the growing demands of the business world. This article addresses PKI at a relatively high-level and does not include details regarding the underlying cryptography.

This article is intended to remove the mystery, fear, and misconceptions of PKI, and offer real world opportunities for its use. Additionally, this article presents business-level reasons for considering a PKI in various environments, and the business problems a PKI can solve. This article is also intended to help organizations determine their requirements and necessity for a PKI, and what features they need for their particular business. This article should be considered as a PKI planning guide.

Why Implement a PKI?

The omnipresence of the Internet and e-commerce technologies present many opportunities, but also pose security and integrity issues. For e-commerce to flourish, businesses, customers, vendors, suppliers, regulatory agencies, and other stakeholders must be assured that trusted business relationships are maintained.

An illustration presents the point. If a merchant today has a physical presence at a store, that is, brick and mortar, and customers patronize them for goods and services, the merchant will typically request and receive payment for these directly from either the customers or their agent (e.g., their bank via the presentation of a monetary instrument such as a check), at the time that the goods and services were bargained for and/or provided. The process of exchanging goods and services for value is almost as universal as the rules by which those conversions take place. In many cases those rules are codified, in others they reflect accepted custom.

Whether systematic or custom, the processes in use today provide for the establishment of a trusted business relationship in that the customer and merchant both authenticate one another to the extent that they are willing to undertake the transaction. If an easily recognized monetary instrument like cash is used for transactions, there may be very little authentication which must occur. If a credit card or check is used, then the authentication may include the establishment of the customer's identity to the merchant. In addition, the authentication may also allow for a measure of non-repudiation to be set so that the customer does not deny the transaction occurred.

This traditional face-to-face transaction requires only minimal interaction and normally does not necessitate the use of other security and integrity mechanisms.

However, for e-commerce on the Internet, additional security and integrity mechanisms become necessary. Merchants are typically not willing to ship goods or perform services until a payment has been accepted for them. In addition, authentication can allow for a measure of non-repudiation so the customer cannot deny the transaction occurred. Similarly, consumers need assurance that they are purchasing from a legitimate enterprise, rather than a hacker’s site whose sole purpose is to collect credit card numbers.

With the changes in today's business environments and the shift from the traditional face-to-face business models, mechanisms must be developed to ensure that trusted relationships are maintained and can flourish.

The implementation of a PKI is intended to provide mechanisms to ensure trusted relationships are established and maintained. The specific security functions in which a PKI can provide foundation are confidentiality, integrity, non-repudiation, and authentication.

Uses of PKI

A PKI does not serve a particular business function; rather, a PKI provides a foundation for other security services. The primary function of a PKI is to allow the distribution and use of public keys and certificates with security and integrity. A PKI is a foundation on which other applications and network security components are built. Systems that often require PKI-based security mechanisms include email, various chip card applications, value exchange with e-commerce (e.g., debit and credit cards), home banking, and electronic postal systems.

A PKI has many uses and applications. As discussed later in this article, a PKI enables the basic security services for such varied systems as:

- SSL, IPsec and HTTPS for communication and transactional security
- S/MIME and PGP for email security
- SET for value exchange
- Identrus for B2B

Some key benefits that PKI and its use of public key cryptography offers for e-commerce and other organizations are as follows:

- Reduces transactional processing expenses
- Reduces and compartmentalizes risk
- Enhances efficiency and performance of systems and networks
- Reduces the complexity of security systems with binary symmetrical methods

In addition, many other similar solutions rely on the fundamentals of public key cryptography such as:

- Student IDs on college campuses
- Voting
- Anonymous value exchange
- Transit ticketing
- Identification (passports and drivers licenses)
- Notarization (contract, emails, etc.)
- Software distribution
- Symmetric key management

Challenges

There are different challenges in the e-commerce world that a well-planned PKI solution addresses; however, there are also many challenges to consider when attempting the selection of a particular PKI solution. Some of these are technical while others are a question of applicability to a specific business model.

It is important to understand that a PKI is not by itself an authentication, authorization, auditing, privacy, or integrity mechanism. Rather, a PKI is an enabling infrastructure that supports these various business and technical needs. In particular, a PKI only allows for the identification of entities. For example, a PKI does not infer trust by itself, but requires the establishment of a trust base, on which the PKI can rely. This requirement means that the basis of trust must be established on a personal, business, or other level, before it can be accepted by the PKI.

A real world example of this is, suppose you misplace your drivers license and are issued a temporary one which does not have your photograph. A temporary license without a picture does not allow a store clerk to determine if you are the owner of it. Therefore, you may not be able to write a check or use a credit card because your identification mechanism, the temporary license, is not acceptable. This indicates that the trust inferred by identification is a rather subjective matter.

The issue of trust often arises when designing a PKI. From an e-commerce standpoint, a notable predicament of remote business transactions is that of original entity authentication. How an organization identifies and authenticates a customer or entity remotely the first time is a difficult problem. The amount of risk that an organization is willing to endure determines the level of effort they must expend during initial authentication. If high-value transactions or transactions with significant legal consequences occur in your organization, a stringent set of tests should be satisfied for a customer or entity to authenticate their identity. Conversely, if there is little risk to issue certificates to entities, for example, userids to access a public Web site, then those tests may be more simple. In any case, the original entity authentication can occur offline and out of band when more stringent means are needed, or dynamically and online for those needing less robust methods. The original entity authentication or initializing problem is not solved directly by a PKI, but must be addressed operationally in each unique business environment.

In the e-commerce environment, this problem is magnified when organizations move from local to regional and then extra-regional environments. How does a clerk in Denmark determine if a driver's license, temporary or otherwise, is legitimate if it was issued in Japan? How do they determine if they should trust the credentials presented? What mechanism do they use to make that determination? How did the original authority, which issues the credentials, determine the identity of the requestor? Do you trust the original authority to perform its identification tests properly? These are all fundamental issues that a PKI must contend with.

With the rapid expansion of e-commerce, closed proprietary legacy systems that only support binary transactional relationships are giving way to more open Internet-based systems, that support remote, many-to-many relationships. Different threats exist in these two very different business models and thus different security functions must be employed to address them. Prior to considering how a PKI can support your business venture into e-commerce, you must identify and evaluate your business requirements for the different security mechanisms that a PKI can enable. You must also identify the specific threats that exist in your environment.

Planning a PKI Infrastructure

This section briefly discusses how different business opportunities have different needs, and how these differences should be considered when planning a PKI.

Defining Business Requirements

A short example will illustrate how different business opportunities have different needs. If a business is a news magazine that freely distributes data over the Internet, the primary concern is maintaining the integrity of the data so it cannot be modified without authorization. Implementing a PKI to simply enable data integrity may not be a cost effective expenditure of resources.

On the other hand, if a business is selling products or services over the Internet, implementing a PKI may be in order. For an e-commerce business, the following must be accounted for when planning a PKI:

- Integrity for the posted prices
- Identification and authentication for a potentially large population of customers
- Confidentiality of customer and transaction information
- Non-repudiation for supporting dispute resolution

Implementing a PKI to enable these various security mechanisms can provide an online merchant with a cost effective approach to risk management.

Other considerations for defining business requirements of a PKI include:

- *Careful planning* – Internet-based e-commerce business solutions are often complex, as are the PKI solutions necessary to support them. Take the time to perform a detailed evaluation of your business and technical environments before taking steps to implement a PKI.
- *Interoperability* – Does your current business model require interoperability? With whom? For what purpose? If your PKI requires interoperability, you should determine which of the different standards and protocols you must adhere. Tangentially, most PKI related standards are in the early stages of development and acceptance. ISO, ANSI, IETF, IEEE, and PKCS are a few examples of standards under development for PKI. Because of the competing standards and protocols and the various interpretations that different vendors have of these, it is critical that organizations determine their interoperability needs.
- *Determining a PKI system and vendor* – There are different PKI and cryptographic systems from competing vendors. Several different protocols, certificate formats, and platforms exist. Some investigation is needed to decide which PKI and

vendor is the best for your particular business enterprise. Often a standards compliant solution from one vendor will not integrate with that of another vendor. This may cause problems if you consider a multi-vendor PKI solution.

- *Performance and capacity* – In situations where large amounts of data must be enciphered for confidentiality, public key cryptography may not be suitable because the cryptographic algorithms perform at relatively slow speeds. Symmetric or secret key cryptography is typically used for these applications. Key management is where public key cryptography plays a role in supporting the encryption of large amounts of data for confidentiality. A PKI can be established for the distribution of the symmetric or secret keys that are subsequently used for the encipherment of data. Public keys and public key certificates can also be significantly larger than symmetric keys and this can affect how they are stored. For example, in the limited memory constraints of a chip card, size can matter.

Structure and Components of a PKI

This section describes the framework of a PKI and how the components of a PKI work together. In addition, this section defines some common terms used in a PKI.

PKI Framework

The framework of a PKI consists of security and operational policies, security services, and interoperability protocols supporting the use of public-key cryptography for the management of keys and certificates. The generation, distribution, and management of public keys and associated certificates normally occur through the use of Certification Authorities (CAs), Registration Authorities (RAs), and directory services, which can be used to establish a hierarchy or chain of trust. CA, RA, and directory services allow for the implementation of digital certificates that can be used to identify different entities. The purpose of a PKI framework is to enable and support the secured exchange of data, credentials, and value (such as monetary instruments) in various environments that are typically insecure, such as the Internet.

A PKI enables the establishment of a trust hierarchy. This is one of the primary principles of a PKI. In Internet-based e-commerce, formal trust mechanisms must exist to provide risk management controls. The concept of trust, relative to a PKI, can be explained by the role of the CA. In the Internet environment, entities unknown to each other do not have sufficient trust established between them to perform business, contractual, legal, or other types of transactions. The implementation of a PKI using a CA provides this trust.

In short, a CA functions as follows. Entities that are unknown to one another, each individually establish a trust relationship with a CA. The CA performs some level of entity authentication, according to its established rules as noted in its Certificate Practices Statement or CPS, and then issues each individual a digital certificate. That certificate is signed by the CA and thus vouches for the identity of the individuals. Unknown individuals can now use their certificates to establish trust between them because they trust the CA to have performed an appropriate entity authentication, and the CA's signing of the certificates attests to this fact. A major benefit of a PKI is the establishment of a trust hierarchy because this scales well in heterogeneous network environments.

Trust Models

The implementation of a PKI requires an analysis of business objectives and the trust relationships that exist in their environment. The awareness of these trust relationships leads to the establishment of an overall trust model that the PKI enforces. The following three common examples of trust models are presented for comparison purposes.

Hierarchical

A hierarchical trust model represents the most typical implementation of a PKI. In its most simple instantiation, this trust model allows end entities' certificates to be signed by a single CA. In this trust model, the hierarchy consists of a series of CAs that are arranged based on a predetermined set of rules and conventions.

For example, in the financial services world, rather than have a single authority sign all end entities' certificates, there may be one CA at a national level that signs the certificates of particular financial institutions. Then each institution would itself be a CA that signs the certificates of their individual account holders. Within a hierarchical trust model there is a trust point for each certificate issued. In this case, the trust point for the financial institution's certificate is the national or root CA. The trust point for an individual account holder is their institution's CA. This approach allows for an extensible, efficient, and scalable PKI.

There are trade-offs to be considered when determining the placement of trust points for end entities in a PKI. In a tiered hierarchy with multiple CAs, compartmentalization of risk can be established, but each CA multiplies the administrative effort necessary to maintain the entire hierarchy. Conversely, a flat hierarchy with a single CA is much easier to administer; however, a failure of that single CA will corrupt the entire trust model and potentially all certificates signed by it.

Distributed (Web of Trust)

A distributed Web of trust is one that does not incorporate a CA. No trusted third party actually vouches for the identity or integrity of any end entity. Pretty Good Privacy (PGP) uses this type of trust model in email environments. This trust model does not scale well into the Internet-based e-commerce world because each end entity is left to its own devices to determine the level of trust that it will accept from other entities.

Direct (Peer to Peer)

Direct peer-to-peer trust models are used with secret or symmetric key-based systems. A trusted third party does not exist in a direct trust model. Thus, each end entity in a peer-to-peer relationship establishes trust with every other entity on an individual basis. This of course, is rather labor-intensive and similar to the Web of trust model. This trust model does not scale well into the Internet-based e-commerce world.

Cross Certification

If PKIs are to be implemented in any widespread fashion, cross certification is another important factor to consider. Instead of using a single global CA, cross certification allows end entities to use a CA based on their particular needs. It is possible that end entities under one CA may need to authenticate end entities under another CA; however, cross certification supports this relatively straightforward process. Essentially, what occurs in a cross certification is that one CA certifies another. As with the generation of an end entity's digital certificate, a CA performs various due diligence tests on the CA it will cross certify. These tests are taken in accordance with the published Certificate Policy and Certificate Practices Statement of the certifying CA.

When a cross-certificate is issued, it extends the trust relationship of a CA. A relying entity, for example, may desire to validate the public key certificate of an end entity whose signing CA’s public key it is not aware of. Assuming that the relying entity trusts its own CA, when it sees a cross-certificate signed by that CA, it will then also trust that other CA, and subsequent certificates signed by it.

The net effect of cross certification is to allow many PKI deployments to be both extensible and scalable.

Security Services

The principle business objectives and risk management controls that can be implemented by a PKI are summarized in this section. An organization should only consider the implementation of a PKI if they have an actual business need for one or more of the security services described in the following sections. Note that these security services depend on the correct use of accepted certificate formats and signing protocols. Without adherence to accepted certificate formats and signing protocols, relying entities cannot determine the correctness of results from various operations.

Confidentiality

Confidentiality means ensuring that the secrecy and privacy of data is provided with cryptographic encryption mechanisms. Customer personal information and legal or contractual data are prime examples of data that should be kept secret by using confidentiality mechanisms. Encryption of data is possible by using either public (asymmetric), or secret (symmetric) cryptography. Since public key cryptography is not as efficient as secret key cryptography for data encipherment, it is normally used to encipher relatively small data objects such as secret keys used by symmetric-based encryption systems. Symmetric cryptographic systems are often incorporated into PKIs for bulk data encryption; thus, they are normally the actual mechanism used to provide confidentiality.

Integrity

Integrity means ensuring that data cannot be corrupted or modified and transactions cannot be altered. Public key certificates and digital signature envelopes are good examples of information that must have an assurance of integrity. Often, the content of messages, emails, purchase transactions and contracts, and information that others rely on, also require the assurance of integrity. Integrity can be provided within a PKI by the use of either public (asymmetric), or secret (symmetric) cryptography. An example of secret key cryptography used for integrity is DES in Cipher Block Chaining mode where a Message Authentication Code (MAC) is generated. Note that in the PKI environment, using symmetric cryptographic systems for implementing integrity does not scale particularly well. Public key cryptography is typically used in conjunction with a hashing algorithm such as SHA-1 or MD5 to provide integrity. A well-designed PKI will use protocols that require the use of these algorithms to provide an efficient integrity mechanism.

Authentication

Authentication means verifying that the identity of entities is provided by the use of public key certificates and digital signature envelopes. Authentication in the e-commerce environment is performed very well by public key cryptographic systems incorporated into PKIs. In fact, the primary goal of authentication in a PKI is to support the remote and unambiguous authentication between entities unknown to each other, using public key certificates and CA trust hierarchies. Authentication in a PKI environment, relies on the mathematical relationship between the public and private keys. Messages signed by one entity can be tested by any relying entity. The relying entity can be confident that only the owner of the private key originated the message, because only the owner has access to the private key.

Non-Repudiation

Non-repudiation means ensuring that data, cannot be renounced or a transaction denied. This is provided through public key cryptography by digital signing. Non-repudiation is a critical security service of any e-commerce application where value exchange, legal, or contractual obligations are negotiated. Non-repudiation is a by-product of using public key cryptography. When data is cryptographically signed using the private key of a key pair, anyone who has access to the public key of that pair can determine that only the owner of the key pair itself could have signed the data in question. For this reason, it is paramount that end entities secure and protect their private keys used for digitally signing data.

PKI Logical Components

Different logical components comprise a PKI. As noted previously, a PKI is a framework of people, processes, policies, protocols, hardware, and software used to generate, manage, store, deploy, and revoke public key certificates. The following outlines the typical logical components in a PKI used in an e-commerce environment. Note that although the security and integrity of the physical infrastructure is important to the successful implementation of a PKI, that subject is beyond the scope of this article. This section discusses the following logical components of a PKI:

- End entities or subscribers
- Certificate authorities
- Certificate policies
- Certificate practices statement
- Hardware security modules

- Public key certificates
- Certificate extensions
- Registration authorities
- Certificate depositories

End Entities or Subscribers

An end entity or subscriber is any user or thing, including inanimate objects, such as computers that have a need for a digital certificate to identify them for some reason. The end entity normally must have the capacity to generate a public/private key pair and some means of securely storing and using the private key. By definition, an end entity is not a CA.

Certificate Authorities

As noted previously in the discussion on Trust, a Certificate Authority plays a critical role in a PKI. According to the IETF, a CA is “an authority trusted by one or more users to create and assign public key certificates.” [Internet X.509 Public Key Infrastructure PKIX Roadmap, March 10, 2000]

To elaborate, a CA functions as a trusted third party and provides various key management services. A CA essentially certifies the identity of an end entity. This is accomplished by an entity providing sufficient proof of their identity to the CA, and then having the CA generate a message containing the entity’s identity and public key. This message is called a certificate and is cryptographically signed by the CA. The level of trust that a CA has depends on the level of acceptance that other entities have in that CA. This level of acceptance depends on the policies and procedures the CA has established to ascertain the user’s identity.

A CA’s public keys must be distributed to all entities that trust the CA’s certificates. If a CA is a Root CA, that is, at the top of the trust hierarchy and has no superior CA to vouch for it, then the CA must distribute its public keys as self-signed certificates with an acceptable key certificate format and distribution protocol. The CA must also make its cleartext public keys available, so that relying entities can resolve the self-signed certificates. The key management-related functions performed by a CA are:

- Certificate generation
- Certificate revocation

A CA works within the context of an overall business policy known as a Certificate Policy (CP) and functions operationally according to a Certificate Practices Statement (CPS). The CP and CPS are discussed briefly in the next section.

Certificate Policy

A primary tenet of e-commerce security is the CP statement. The CP statement provides the overall guiding principles that an organization endorses regarding who may do what and how to systems and data. A CP also specifies how controls are managed. In addition, a CP names a set of rules that indicates the applicability of a public key certificate to a particular community or class of applications with common security requirements. For example, a particular CP might indicate applicability of a type of public key certificate to the authentication of electronic data interchange transactions for the trading of goods for monetary value.

Each PKI implementation should reflect the following in a CP statement:

- Purpose of the PKI
- Specific business requirements the PKI addresses through:
  - Security architecture
  - Associated trust model and threat profile
  - Specific security services the PKI supports

The CP statement should also reflect the specific business realities of a particular customer. For example, e-commerce merchants may desire assurance of timely payment for goods and services rendered through the use of authentication and non-repudiation mechanisms. Alternately, a facilitator of proprietary email services may only require confidentiality of information through the use of encryption facilities.

Certificate Practices Statement

The details of a policy statement should be published in a Certificate Practices Statement or CPS. The CPS is a statement of the practices that a CA employs in issuing public key certificates. The CPS document enumerates the procedural and operational practices of a PKI. The CPS should detail all processes within the life cycle of a public key certificate including its generation, issuance, management, storage, deployment, and revocation. The CPS should also specify the original entity authentication process that an end entity must be validated through before participating in a PKI. The objective of the CPS is to instill trust in the PKI such that the user community at large will have sufficient confidence to participate in it.

Hardware Security Modules

Hardware Security Modules (HSMs) are another primary component of a CA. A CA must instill trust in not only its client base but also in those who rely upon the certificates issued to subscribers. Since that trust is predicated upon the security and integrity of the CA's private keys used to sign the public key certificates of subscribers, it is necessary that those private keys be secured as best as possible. For this reason, CAs should only store and use their private keys in specialized computer equipment known as HSMs. HSMs are also known as Tamper Resistant Security Modules or (TRSMs) and because of their reliance in the financial services industry, TRSMs are well-defined through various acceptance standards in terms of their features and operational characteristics. The implementation and use of a qualified HSM is critical to any CA and the PKI it supports. Various standards are used to categorize HSMs, for example, FIPS-140-1.

Public Key Certificates

As stated previously, a CA’s primary purpose is to support the generation, management, storage, deployment, and revocation of public key certificates. A public key certificate demonstrates or attests to the binding of an end entity’s identity and its public key. That is, it contains enough information for some other relying entity to validate and verify the identity of the owner of the certificate. The basic constructs of a certificate should include the name of an entity, identifying information about the entity, expiration period for the certificate, and the entity’s public key. Normally one would expect a certificate used in the e-commerce world to include additional information and be based upon an accepted format to allow interoperability. Other additional and useful information may be included in a certificate: serial numbers, the CA's name, the CA's public key certificate itself, the type of algorithms used to generate and verify the keys and certificate, and any other information that the CA generating the certificate considers useful.

The most widely used format for digital certificates are those based on the IETF X.509 standards. A detailed semantic profile of X.509 based public key certificates can be found in the IETF RFC 2459 and related documentation. It should be noted that there is no one single definition of a public key certificate defined in the IETF standards. Vendors, integrators, and others offering PKI solutions each have their own ideas as to what extensions and particular data an X.509 certificate should contain. Organizations should evaluate their business needs relative to the constructs of the public key certificates that they wish to issue.

Certificate Extensions

Certificate extensions provide additional information within a certificate and allow them to be tailored for the particular needs of an organization. Be aware that certificate extensions can affect the interoperability of certificates if a relying party does not recognize the structure or content of the certificate extensions. The types of information that may be found in a certificate extension include: policy, usage, revocation, and naming data, which provide particular details unique to an organization's PKI.

Registration Authorities

A Registration Authority (RA) is an optional but common component of a PKI. An RA is used to perform some of the administrative tasks that a CA would normally undertake. Most importantly, an RA is delegated, with the CA's explicit permission, the authority to perform tasks on behalf of the CA. The primary purpose of an RA is to verify an end entity's identity and determine if an end entity is entitled to have a public key certificate issued. The RA must enforce all policies and procedures defined in the CA's CP and CPS. A typical function of an RA is to interrogate an end entity's certificate request by examining the name, validity dates, applicable constraints, public key, certificate extensions, and related information. The RA may also be responsible for performing due diligence tests on the end entity. This responsibility may be as simple as ensuring the name of the end entity is unique within the scope of the PKI; or, it may be as involved as making credit checks on potential clients.

Certificate Depositories

As with an RA, a certificate depository, sometimes referred to as a certificate directory, is also an optional but common component of a PKI. A certificate depository may be an efficient solution for closed systems (e.g., intranet) or those in isolated processing environments (e.g., chipcard-based applications) where the Root CA public key is distributed locally or revocation lists are stored locally. Many other situations warrant the use of a certificate or Certificate Revocation List (CRL) depository. The use of depositories for CRL distribution is discussed later in this article.

Certificate distribution can be accomplished by simply publishing certificates in a directory controlled by a CA or RA. When the directory is controlled by the CA or RA, the certificate distribution process is greatly simplified. Rather than trying to distribute every certificate to a unique point, the CA simply updates the directory. A critical factor is that only the CA must have the authority to update or modify the directory, but the directory must be publicly readable.

LDAP is a good example of a simple and efficient standards based directory format and protocol that can be used for certificate distribution. In particular, LDAP is optimized for easy read access by thin clients that are a necessity in many PKI implementations. Because LDAP defines an API for the application layer used by different clients, users realize a level of scalability and portability, and do not have to contend with the upper layers of the stack. This API also makes it easier for different applications to quickly implement LDAP.

Although LDAP is becoming a de facto standard for certificate distribution, organizations that need an even more robust certificate depository can consider alternatives such as X.500.

PKI Functions

This section discusses the following basic processes which are common to all PKIs:

- *Public key cryptography* – Includes the generation, distribution, administration, and control of cryptographic keys.
- *Certificate issuance* – Binds a public-key to an individual, organization, or other entity, or to some other data—for example, an email or purchase order.
- *Certificate validation* – Verifies that a trust relationship or binding exists and that a certificate is still valid for specific operations.
- *Certificate revocation* – Cancels a previously issued certificate and either publishes the cancellation to a Certificate Revocation List or enables an Online Certificate Status Protocol process.

Public Key Cryptography

This section is a brief overview of the cryptography that is incorporated into a PKI. Current public key cryptography as described in this article is mostly attributed to Diffie and Hellman and Rivest, Shamir and Adleman.

Because of its widespread use in e-commerce, this article focuses on the RSA (named for its creators: Rivest, Shamir and Adleman) public key cryptographic system. RSA is a public key cryptographic algorithm that is based on the hard mathematical problem of factoring composite numbers. The keys used by the RSA crypto-system are based on the product of two large prime numbers that derive their cryptographic strength from the fact that it is difficult to factor large composite numbers of this kind. RSA uses a pair of keys: a public key which is made known to many entities, and a private key for which secrecy and integrity are strictly controlled and only used by the owner of that key. Given an appropriate key length, it is

computationally infeasible to determine one key from another. The basic cryptographic feature of RSA is that it allows the encryption of clear text data with one key but decryption with the other. This basic cryptographic feature is what provides RSA its asymmetry.

RSA performs the generation of a public/private key pair as follows:

Two large primes, *p* and *q* are used to compute their product $n = pq$, where *n* is called the modulus. A number is chosen, *e*, which is less than *n* and relatively prime to $(p-1)(q-1)$, which means *e* and $(p-1)(q-1)$ have no common factors except 1. Another number is chosen, *d*, such that $(ed - 1)$ is divisible by $(p-1)(q-1)$. This is the inverse of *e* and means that $ed = 1 \bmod (p-1)(q-1)$.

The values *e* and *d* are called the public and private exponents, respectively. The public key is the pair (n, e) and the private key is (d).

RSA supports two basic modes of operation: encryption and digital signatures. These are outlined in the following sections.

Key strength

Key strength is a critical factor in cryptography. An insufficiently strong key will corrupt even the best-designed PKI. The level of cryptographic strength of any key depends on its use, but in general, the following apply:

- To prevent their potential compromise, keys must be replaced prior to the time feasible to determine them through cryptanalysis.
- A key must be replaced by a new key within the time feasible to perform a successful dictionary or brute force attack on the old key.
- A key must not be used to ensure the security or integrity of data where it is feasible to determine that key in a period of time less than the useful life of that data.

Note that these points do not dictate specific key lengths, but rather provide guidance on how to determine if a key is fit for purpose.

Encryption

RSA encryption is the creation of ciphertext by one entity using another entity's public key to perform the encryption. This allows many entities to send one entity encrypted messages without the need to first exchange secret or private cryptographic keys. And because the encryption was performed with a public key, it can only be decrypted with the corresponding private key so that only the intended recipient can decrypt and read the original message.

RSA encipherment is performed as follows:

$c = m^e \bmod n$

Where *m* is the message to be enciphered and *c* is the resultant ciphertext.

The specific operation performed is the exponentiation of $m^e \bmod n$, where *e* and *n* are the public key of the recipient of the ciphertext.

The recovery of the ciphertext by the recipient occurs as follows:

$m = c^d \bmod n$

The specific operation performed is the exponentiation of $c^d \bmod n$ where *d* and *n* are the recipients private key.

Digital Signatures

RSA digital signatures work in an inverse fashion from encryption. Digital signatures are generated by performing the encryption of some cleartext using one's own private key. This encryption allows one entity to send a message to many other entities that may then authenticate that message, without the need to first exchange secret or private cryptographic keys. The recipient simply decrypts the message with the originator's public key.

An RSA digital signature is performed as follows:

$s = m^d \bmod n$

Where *m* is the message and *s* the resultant digital signature.

The specific operation performed is the exponentiation of $m^d \bmod n$ where *d* and *n* are the sender's (of the message) private key.

The authentication of the digital signature by the recipient occurs as follows:

$m = s^e \bmod n.$

The specific operation performed is the exponentiation of $s^e \bmod n$ where *e* and *n* are the sender's public key.

Hash Functions

Given its general acceptance and incorporation in cryptographic standards, SHA-1 is a good example of a hash function. The purpose of SHA-1 is to provide a non-secret key-based hash. It is widely used in public key cryptography, especially in conjunction with RSA, where public key-based authentication mechanisms are required.

SHA-1 is a one-way hash function that takes an arbitrary length message, processes it, and returns a fixed length, 20 byte value. A secret key is not used. The primary characteristic of SHA-1 is that it provides a mechanism that makes it easy to compute a hash from some data, but difficult to determine any data from a computed hash value.

There are two important security properties of SHA-1:

- SHA-1 is one-way, given SHA-1(x) it is difficult to find x.
- SHA-1 is collision-resistant, given it is hard to find x,y with $x \neq y$ satisfying: SHA-1(x)=SHA-1(y).

SHA-1 offers the benefit of providing a unique image or value of a message that has been hashed without the necessity for performing a more time consuming public key encryption of the entire message. When this unique value is sent with the message which was used to create it, the receiver is provided with the ability to test the message to determine if it has been altered or not.

Symmetric Key Encryption

An overview of public key cryptography is not complete without mentioning symmetric or secret key cryptography. The reason for this is that public key cryptographic systems lack the positive performance characteristics of symmetric systems. The ramification of this under-performance is that, in practice, symmetric systems such as DES, Triple-DES, or IDEA are used to perform encipherment of data for confidentiality, where the data is more than a negligible length. In this way, symmetric key encryption supplements public key cryptographic systems.

The fundamental characteristics of symmetric cryptographic systems used with PKI are that the same key is used for both encryption and decryption. Because of this symmetry the key must be kept secret and shared only between two parties.

Key Separation

Although the RSA cryptographic system allows a single pair of keys to be used for both encryption and digital signature operations, this is not a good practice. The concept of key separation is that keys must only be employed for specifically defined functions. This helps to isolate any corruption that may occur from the compromise of a cryptographic key, and to also avoid certain cryptographic attacks. This isolation is referred to as compartmentalization of risk. In a well-designed PKI, to enable an appropriate level of risk compartmentalization, different key pairs should be used for encryption and digital signature generation.

Public Key Certificate Binding

To participate in a PKI, an end entity must enroll or register in a PKI. The result of this process is the generation of a public key certificate. The primary objective of a CA is to bind the identifying information and credentials supplied by an end entity with the public key of that end entity. The binding is declared when a trusted CA digitally signs the public key certificate with its private key. This process is outlined in the following sections.

Entity Authentication

Entity authentication or enrollment can be a difficult problem. The original authentication or initializing process which is used to identify and validate the entity, may be a perfunctory matter or one of extreme due diligence. This depends on the business risk model of an organization and is reflected in their CA. If a sufficient level of validation does not occur during the original identification and authentication of an entity, any subsequent reliance on the credentials and certificates issued to that identity could be suspect.

If there is little negative consequence to the issuance and use of a certificate, for example, renting movies from a video store, there may be only a simple entity authentication process. Conversely, the issuance of a certificate for use by a passport agency may be more robust if that is a nation's first line of defense against cyber-terrorists. As a simple matter, if an organization is attempting to replicate an existing manual process that requires some form of identity validation, for example, to open a savings account at a bank, they should replicate the same process they use today. The process may be automated or manual but the credentials required and the due diligence verification processes should be the same or equivalent.

Note that the information and credentials requested and the enrollment process which a CA or RA uses must be reflected in its published CPS.

Certificate Generation

To generate a certificate, the CA performs the following steps. Note that although an RA may be used, it is excluded from the process here for brevity.

- Acquire a public key from the end entity.
- Verify the identity of the end entity.
- Determine the attributes needed for this certificate, if any.
- Format the certificate.
- Digitally sign the certificate data.

Acquiring the Public Key

Depending on the value proposition and business risks associated with the issuance of a public key certificate, the CA may take basic measures when obtaining the required identification and certificate information. For example, the CA may allow it to be sent electronically over the Internet, or the CA could use more sophisticated means such as mandating out of band manual methods (e.g., using bonded couriers). The type and integrity of the credentials requested by the CA during the enrollment process also depends on the intentions of the CA.

Verify the Identity of the End Entity

Again, depending on the business model in place and the amount of reliance on the public key certificates, the CA may take simple measures to authenticate the end entity. For example, they could simply take the end entity's word that they are who they say they are, or the CA could institute more stringent measures, such as a detailed due diligence process as the business proposition merits. In any case, the most important factor in the binding process is to ensure that an entity’s identity is verified unambiguously.

Formatting the Certificate

Before the certificate is signed by the CA, all data to be placed into the certificate is collected and formatted. The specific data content and format of a public key certificate can vary depending on the needs of the PKI.

Signing the Certificate

Finally, the certificate is digitally signed under the private key of the CA used for signing certificates. Once signed it can be distributed and/or published using different vehicles.

Certificate Validation

The following provides an example of certificate validation using a digital envelope. This is one of several approaches for using digital envelopes, other approaches may be valid as well. Review your specific security and business requirements before deciding on a specific approach to use digital envelopes.

An Example of a PKI in Action

The following provides an example of public key cryptography in use for both confidentiality and integrity. The purpose of this example is to illustrate how public key cryptography mechanisms can be used in a PKI. In this example, both parties, Alice and Bob, share a common trust point; that is, they both use the same CA to have their certificates signed. For this reason, they do not have to evaluate a chain of trust to determine the credibility of any other CA. The example is not necessarily appropriate for every business proposition.

Precursor Steps

1. Alice and Bob each generate a public/private key pair.
2. Alice and Bob each provide their public keys, name, and descriptive information to an RA.
3. The RA validates their credentials and forwards the certificate requests to the CA.
4. The CA generates a certificate for Alice and Bob's public keys by formatting their public keys and other information, and then signs the certificate with the CA's private keys.
5. The results of this operation are that Alice and Bob each have a public/private key pair and a public key certificate.
6. Alice and Bob each generate a secret symmetric key.

Now Alice and Bob each have a public/private key pair, a digital key certificate issued by a common trusted third party (i.e., the CA), and a secret symmetric key.

Suppose now that Alice wishes to send Bob some data that he will rely upon and thus requires an assurance of integrity; that is, the content of the message cannot be altered. Alice and Bob also want the assurance of confidentiality and want no other parties to be able to view the information. A contract for goods and services where parties agree upon price, time, and value exchange is a good example of such a process. The steps taken to perform the transaction are as follows.

In this example, steps 1-5 involve Alice sending data that needs confidentiality and integrity to Bob, using a digital signature. Steps 6-10 involve Bob decrypting the data.

1. Alice takes the message in question, formats it according to the protocol rules she and Bob have agreed to and then hashes the message. The hash provides a unique value for the message and will later be used by Bob to test the validity and integrity of the message.

2. Alice concatenates the message and the hash and then signs (i.e., encrypts) these with her private key. This signing provides message integrity because Bob is assured that only Alice could have generated the signature because only Alice has access to the private key used to sign the message. Note that anyone with access to Alice's public key can recover the signed message and so, we have only established message integrity at this point, but not confidentiality.

3. Because Alice and Bob would like to keep the message confidential, Alice encrypts the signed message and hash with her secret symmetric key. This key is only shared between Alice and Bob and no one else. Note that enabling confidentiality can be implemented through various techniques and protocols. In this example, a symmetric key is used because it can be more efficient than using a public key for encrypting lengthy messages such as the purchase order here.

4. Alice must provide Bob with her secret symmetric key to enable Bob to decrypt the message encrypted with that key. Alice signs (i.e., encrypts) her secret symmetric key using Bob's public key. For this example, we assume Alice obtained Bob's public key previously. This generates what is known as a digital envelope and which only Bob can recover (i.e., decrypt) because only Bob has access to the private key that is needed to recover the digital envelope. Remember, what is done with one key of a pair can only be undone with the other key of the pair. This provides confidentiality over the transmission of Alice's secret symmetric key to Bob.

5. Alice forwards to Bob the original message and the hash that are both encrypted with her secret symmetric key and the digital envelope containing the secret key encrypted with Bob's public key.

FIGURE 1 illustrates Alice using a digital signature to send data to Bob (steps 1-5).

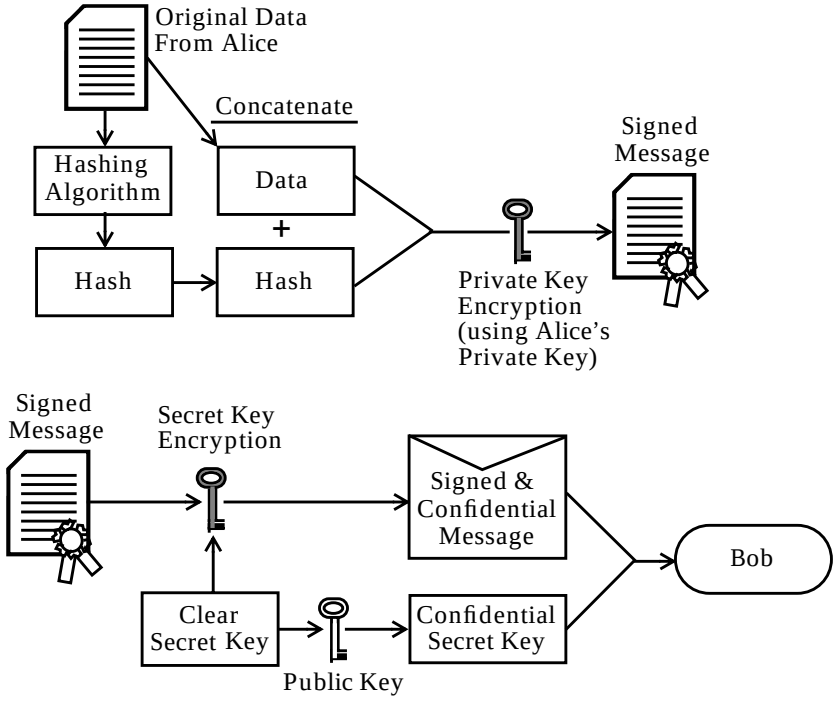

FIGURE 1 Overview of Using a Digital Signature

6. Bob takes the digital envelope he received from Alice and recovers (i.e., decrypts) it with his private key. The results of performing this operation provide Bob with the secret symmetric key that Alice previously used to encrypt the message and the hash of the message.

7. Bob can now recover (i.e., decrypt) the encrypted message and hash using Alice's secret symmetric key. Bob now has the signed cleartext message and the signed hash of it.

8. Bob now recovers (i.e., decrypts) the signed message and hash of the message by using Alice's public key. Remember, what is done with one key of a pair can only be undone with the other key of the pair.

9. To ensure that no modifications have been made to the message, Bob takes the original message and hashes it using the exact same process that Alice used originally.

10. Finally, Bob compares the hash he has just produced with the hash he recovered from the original message. If they match he is assured of the message's integrity.

FIGURE 2 illustrates Bob decrypting the data and comparing hashes (steps 6-10).

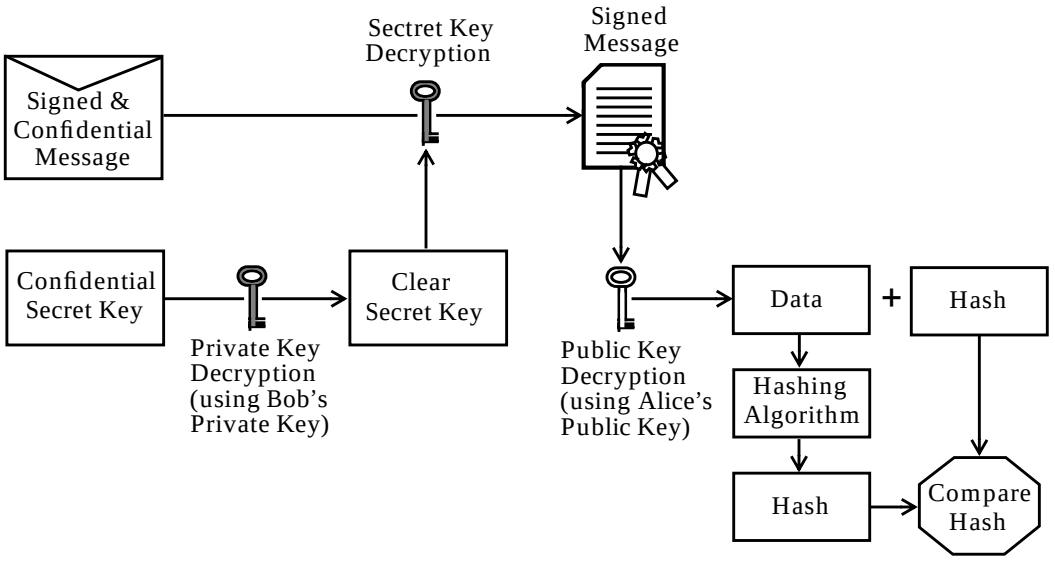

FIGURE 2 Overview of the Decryption and Hash Comparison Process

Certificate Revocation

Although public key certificates are issued for a fixed period of time before they become void, situations can arise where they are no longer trustworthy and thus must be prematurely expired. This is known as certificate revocation. There are many reasons that a certificate should be revoked. The private keys could be compromised, a user is no longer a customer, or there is a change in the information incorporated within a certificate that is used to determine its validity.

Certificate revocation must be initiated by the CA or their delegate such as an RA, which originated the end entity's certificate. The predominant vehicle for certificate revocation is known as a Certificate Revocation List or CRL. A CRL is a list generated by the CA that contains unique information about the revoked certificates which enables relying entities to determine if a certificate is valid or not. A CRL entry should be serialized, time-stamped, and signed by the CA. It is normally the relying entity's responsibility to retrieve the CRL.

A CRL must be published in a publicly available repository or directory. LDAP is the industry's current directory of choice and has been developed as an X.500 compliant protocol for the Internet. LDAP defines the directory query, data storage, and management protocols.

The timely publication of the CRL is critical in Internet and e-commerce environments. This timely publication is particularly critical during the latency period between when a CA revokes a certificate and its subsequent publication where there is a reliance upon the certificate for making critical decisions. If for example, a high-value transaction is being negotiated and a seller is relying on the CRL so they can make the decision to sell or not, a timely CRL publication cycle may be critical to them.

There are other types of CRLs such as, segmented-CRLs, delta-CRLs, and CRL-like mechanisms—for example, the Online Certificate Status Protocol (OCSP). OCSP is the most interesting because it can address the latency issue associated with standard CRL publication. OCSP is a mechanism that actually requests, in real time, a status check for a particular certificate from the originating CA. This has the benefit of not only being timely but in fact, reduces or eliminates the necessity for a CA to publish a CRL. The CRL simply waits for status check requests and responds to them as necessary.

Conclusion

PKI is a complex subject and still evolving in terms of its utilization in the commercial and e-commerce sectors. Although the underlying technology is quite sound, issues exist in areas such as interoperability and performance. Nonetheless, PKI offers considerable benefits to those in need of the basic security services described in this article.

Those considering a PKI should evaluate their environment to understand where they require basic security services and thus where a PKI would be most useful.

When progressing to a PKI, careful planning is critical. Start small with a pilot implementation. This will allow you to gain an understanding of the issues and also the operational, security, and practical aspects particular to your environment.

With the successful implementation of a pilot PKI and a clear understanding of focused goals and objectives, which can realistically be satisfied by a PKI, you can proceed to a more comprehensive implementation of a PKI.

Acronyms

- ANSI – American National Standards Institute
- B2B – Business to Business

- DES - Data Encryption Standard
- FIPS – Federal Information Processing Standard
- HTTPS – Secure Hypertext Transaction Protocol
- IEEE – Institute of Electrical and Electronic Engineers
- IETF – Internet Engineering Task Force
- IPsec – Secure Internet Protocol
- ISO – International Organization for Standardization
- PGP – Pretty Good Privacy
- PKCS – Public Key Cryptography Standards
- PKI - Public Key Infrastructure
- SET – Secure Electronic Transactions
- SHA-1: Secure Hash Standard
- S/MIME – Secure Multipurpose INternet Mail Extensions
- SSL – Secure Socket Layer

Bibliography

Carlisle Adams, *Understanding Public-key Infrastructure*, Macmillan Technical Publishing, November 1999

Tom Austin, *PKI - A Wiley Tech Brief*, Wiley, John & Sons Incorporated, December 2000

Arto Salomaa, *Public-Key Cryptography*, Springer-Verlag, Berlin, Heidelberg, 1990

Bruce Schneier, *Applied Cryptography*, Wiley, John & Sons, Incorporated, October 1995

Alfred Menezes, *Handbook of Applied Cryptography*, CRC Press, LLC, October 1996

PKI Practices and Policy Framework Draft, ANSI X9.79 standard

William Polk, *Bridge Certification Authorities: Connecting B2B Public Key Infrastructures*, 2000

NIST PKI Project Team, *Certificate Issuing and Management Components Protection Profile*, 2001

Russell Housley, *Internet Public Key Infrastructure*, X509 Certificates and CRL Profile RFC 2459, 1999

S. Chokhani, *Certificate Policy and Certification Practices Framework*, RFC 2527

Author's Bio: Joel Weise

Joel Weise has worked in the field of data security for over 20 years. As a Senior Security Architect for Sun Professional Services, he designs system and application security solutions for a range of different enterprises from financial institutions to government agencies. He specializes in cryptography and public key infrastructures. Prior to joining Sun, Joel was a Senior Project Manager for Visa International. There he was responsible for developing cryptographic standards, designing key management and cryptographic systems and architecting security solutions for chipcard, Internet, and other new products.