Public Key Infrastructure

05/30/2018 • 3 minutes to read •

Public-key cryptography (also called asymmetric-key cryptography) uses a key pair to encrypt and decrypt content. The key pair consists of one public and one private key that are mathematically related. An individual who intends to communicate securely with others can distribute the *public key* but must keep the *private key* secret. Content encrypted by using one of the keys can be decrypted by using the other. Assume, for example, that Bob wants to send a secure email message to Alice. This can be accomplished in the following manner:

1. Both Bob and Alice have their own key pairs. They have kept their private keys securely to themselves and have sent their public keys directly to each other.
2. Bob uses Alice's public key to encrypt the message and sends it to her.
3. Alice uses her private key to decrypt the message.

This simplified example highlights at least one obvious concern Bob must have about the public key he used to encrypt the message. That is, he cannot know with certainty that the key he used for encryption actually belonged to Alice. It is possible that another party monitoring the communication channel between Bob and Alice substituted a different key.

The public key infrastructure concept has evolved to help address this problem and others. A public key infrastructure (PKI) consists of software and hardware elements that a trusted third party can use to establish the integrity and ownership of a public key. The trusted party, called a *certification authority* (CA), typically accomplishes this by issuing signed (encrypted) binary certificates that affirm the identity of the certificate subject and bind that identity to the public key contained in the certificate. The CA signs the certificate by using its private key. It issues the corresponding public key to all interested parties in a self-signed CA certificate. When a CA is used, the preceding example can be modified in the following manner:

1. Assume that the CA has issued a signed digital certificate that contains its public key. The CA self-signs this certificate by using the private key that corresponds to the public key in the certificate.
2. Alice and Bob agree to use the CA to verify their identities.
3. Alice requests a public key certificate from the CA.
4. The CA verifies her identity, computes a hash of the content that will make up her certificate, signs the hash by using the private key that corresponds to the public key in the published CA certificate, creates a new certificate by concatenating the certificate content and the signed hash, and makes the new certificate publicly available.
5. Bob retrieves the certificate, decrypts the signed hash by using the public key of the CA, computes a new hash of the certificate content, and compares the two hashes. If the hashes match, the signature is verified and Bob can assume that the public key in the certificate does indeed belong to Alice.
6. Bob uses Alice's verified public key to encrypt a message to her.
7. Alice uses her private key to decrypt the message from Bob.

In summary, the certificate signing process enables Bob to verify that the public key was not tampered with or corrupted during transit. Before issuing a certificate, the CA hashes the contents, signs (encrypts) the hash by using its own private key, and includes the encrypted hash in the issued certificate. Bob verifies the certificate contents by decrypting the hash with the CA public key, performing a separate hash of the certificate contents, and comparing the two hashes. If they match, Bob can be reasonably certain that the certificate and the public key it contains have not been altered.

A typical PKI consists of the following elements.

Element	Description
Certification Authority	Acts as the root of trust in a public key infrastructure and provides services that authenticate the identity of individuals, computers, and other entities in a network.
Registration Authority	Is certified by a root CA to issue certificates for specific uses permitted by the root. In a Microsoft PKI, a registration authority (RA) is usually called a subordinate CA.
Certificate Database	Saves certificate requests and issued and revoked certificates and certificate requests on the CA or RA.
Certificate Store	Saves issued certificates and pending or rejected certificate requests on the local computer.
Key Archival Server	Saves encrypted private keys in the certificate database for recovery after loss.

The Certificate Enrollment API enables you to submit certificate and key archival requests to certification and registration authorities and install the issued certificate on a local computer. It does not enable you to directly manipulate the certificate database or certificate store.

The following topics discuss the Microsoft public key infrastructure in more detail:

- X.509 Public Key Certificates
- PKI Elements
- Trust Models

Related topics

About the Certificate Enrollment API

Is this page helpful?

Yes No

In this article

Related topics