

Public Key Infrastructure (PKI): Theory and Practice

Jorge Martinez de Salinas
University of the Basque Country
jorge.marsal@gmail.com

June 11, 2009

Abstract

The article highlights the most important concepts regarding Public Key Infrastructure (PKI). It also includes a practical step-by-step guide explaining how to set up a PKI on Linux using the OpenSSL package.

Contents

1	PKI	3
1.1	Theory	3
1.2	Practice	8

List of Figures

1	CA and client certs.	9
2	Public key and CA's sign in the client's cert.	10

1 PKI

1.1 Theory

Limitations of Conventional Secret-Key Cryptography

The solution to problems of identification, authentication, and privacy in computer-based systems lies in the field of cryptography. Because of the non-physical nature of the medium, traditional methods of physically marking the media with a seal or signature (for various business and legal purposes) are useless. Rather, some mark must be coded into the information itself in order to identify the source, authenticate the contents, and provide privacy against eavesdroppers.

Privacy protection using a symmetric algorithm, such as that within DES (the government-sponsored Data Encryption Standard) is relatively easy in small networks, requiring the exchange of secret encryption keys among each party. As a network proliferates, the secure exchange of secret keys becomes increasingly expensive and unwieldy. Consequently, this solution alone is impractical for even moderately large networks.

DES has an additional drawback, it requires sharing of a secret key. Each person must trust the other to guard the pair's secret key, and reveal it to no one. Since the user must have a different key for every person they communicate with, they must trust each and every person with one of their secret keys. This means that in practical implementations, secure communication can only take place between people with some kind of prior relationship, be it personal or professional.

Fundamental issues that are not addressed by DES are authentication and non-repudiation. Shared secret keys prevent either party from proving what the other may have done. Either can surreptitiously modify data and be assured that a third party would be unable to identify the culprit. The same key that makes it possible to communicate securely could be used to create forgeries in the other user's name.

A Better Way: Public Key Cryptography

The problems of authentication and large network privacy protection were addressed theoretically in 1976 by Whitfield Diffie and Martin Hellman when they published their concepts for a method of exchanging secret messages without ex-

changing secret keys. The idea came to fruition in 1977 with the invention of the RSA Public Key Cryptosystem by Ronald Rivest, Adi Shamir, and Len Adleman, then professors at the Massachusetts Institute of Technology.

Rather than using the same key to both encrypt and decrypt the data, the RSA system uses a matched pair of encryption and decryption keys. Each key performs a one-way transformation upon the data. Each key is the inverse function of the other; what one does, only the other can undo.

The RSA Public Key is made publicly available by its owner, while the RSA Private Key is kept secret. To send a private message, an author scrambles the message with the intended recipient's Public Key. Once so encrypted, the message can only be decoded with the recipient's Private Key.

Inversely, the user can also scramble data using their Private Key; in other words, RSA keys work in either direction. This provides the basis for the "digital signature," for if the user can unscramble a message with someone's Public Key, the other user must have used their Private Key to scramble it in the first place. Since only the owner can utilise their own private key, the scrambled message becomes a kind of electronic signature – a document that nobody else can produce.

Advantages of the PKI Approach

The PKI approach to security does not take the place of all other security technologies; rather, it is an alternative means of achieving security. The following advantages of PKI have led to its emergence as an industry standard for securing Internet and e-commerce applications.

- PKI is a standards-based technology.
- It allows the choice of trust provider.
- It is highly scalable. Users maintain their own certificates, and certificate authentication involves exchange of data between client and server only. This means that no third party authentication server needs to be online. There is thus no limit to the number of users who can be supported using PKI.
- PKI allows delegated trust. That is, a user who has obtained a certificate from a recognized and trusted certificate authority can authenticate himself to a server the very first time he connects to that server, without having previously been registered with the system.

PKI is emerging as the foundation for secure electronic commerce and Internet security by providing the cornerstones of security (i.e., authentication, encryption and non-repudiation).

Authentication: The importance of authentication, verifying the identity of users and machines, becomes crucial when an organization opens its doors to the Internet. Strong authentication mechanisms ensure that persons and machines are the entities they claim to be.

Encryption: Encryption algorithms are used to secure communications and ensure the privacy of data sent from one computer to another.

Non-repudiation: PKI can be used to provide non-repudiation through digital signatures. This proves that a specific user performed certain operations at a given time.

Together, these elements combine to provide a secure, non-breakable environment for deploying e-commerce and a reliable environment for building virtually any type of electronic transactions, from corporate intranets to Internet-based eBusiness applications.

Components of PKI

The main components of a public key infrastructure are:

- **Digital certificates:** Digital "identities" issued by trusted third parties, that identify users and machines. They may be securely stored in wallets or in directories.
- **Public and private keys:** Form the basis of a PKI for secure communications, based on a secret private key and a mathematically related public key
- **Secure sockets layer (SSL):** An Internet-standard secure protocol
- **Certificate Authority (CA):** Acts as a trusted, independent provider of digital certificates

Other important factors that enable the deployment of PKI include: secure storage of certificates and keys; management tools to request certificates, access wallets and administer users; and a directory service acting as a centralized repository for certificates.

Secure Credentials: Certificate-Based Authentication in PKI

Establishing user identity is of primary concern in distributed environments; otherwise, there can be little confidence in limiting privileges by user. Passwords are the most common authentication method in use, but for particularly sensitive data, you need to employ stronger authentication services.

Having a central facility authenticate all members of the network (clients to servers, servers to servers, users to both clients and servers) is one effective way to address the threat of nodes on a network falsifying their identities. This method involves certificates and certificate authorities.

Certificate Authorities

A certificate authority (CA) is a trusted third party that certifies that other entities—users, databases, administrators, clients, servers—are who they say they are. When it certifies a user, the certificate authority verifies the user’s identity and grants a certificate, signing it with the certificate authority’s private key. The certificate authority has its own certificate and public key, which it publishes, as well as a private key, which is securely maintained. Servers and clients use the CA’s root certificate to verify signatures that the certificate authority has made. A certificate authority might be an external company that offers certificate services, or an internal organization such as a corporate MIS department

Certificates

A certificate is like an electronic passport that proves the identity of a user or device that seeks to access the network. The certificate ensures that the entity’s information is correct and that the public key actually belongs to that entity. A certificate is created when an entity’s public key is signed by a trusted identity (a certificate authority). It contains information such as the following:

- the certificate user’s name
- an expiration date
- a unique serial number assigned to the certificate by the CA
- the user’s public key
- information about the rights and uses associated with the certificate
- the name of the certificate authority that issued the certificate
- the CA’s signature
- an algorithm identifier that identifies which algorithm was used to sign the certificate

A trusted certificate, sometimes known as a root key certificate, typically belongs to a third party entity that is trusted to issue certificates. It is obtained in a secure manner and, operationally, does not need to be validated for its authenticity each time it is accessed because it is self-signed. A client or a server can validate that an entity is who it claims to be by verifying that the entity’s certificate was issued by a known and trusted certificate authority.

Typically, certificate authorities whom you trust issue the user certificates. Oracle provides several default trusted certificates, so users do not have to install their own. These trusted certificates also enable servers to perform SSL authentication to clients who have wallets containing only trusted certificates.

Clients and servers use these credentials to access secure services, such as SSL, using public key cryptography. A wallet also represents a storage facility that is location- and type-transparent once it is opened.

Authentication Methods Used with PKI

Popular authentication methods used with PKI include:

- Secure Sockets Layer Authentication and X.509v3 Digital Certificates
- Entrust/PKI Authentication

Secure Sockets Layer Authentication and X.509v3 Digital Certificates

The Secure Sockets Layer (SSL) is an industry standard protocol that provides authentication, data encryption, and data integrity, in a public-key infrastructure. SSL is widely employed over the Internet to give users established digital identities and to prevent eavesdropping, tampering with, or forging messages.

SSL provides authentication through the exchange of certificates that are verified by trusted certificate authorities. SSL uses digital certificates (X.509 v3), and a public/private key pair to authenticate users and systems.

The most widely used public key certificates comply with the X.509 format, and the X.509 Version 3 certificate is the current industry standard format. A public key infrastructure relies on X.509 certificates, also called digital certificates, or public-key certificates, for public-key authentication.

X.509v3 digital certificates contain the following:

- The certificate owner's Distinguished Name (DN), which uniquely identifies the owner
- The Distinguished Name of the certificate issuer, which uniquely identifies the certificate authority
- The certificate owner's public key
- The issuer's signature
- The dates for which the certificate is valid
- The serial number of the certificate

The SSL protocol has gained the confidence of users, and it is perhaps the most widely-deployed and well-understood encryption protocol in use today.

1.2 Practice

Now, we can put the theoretical knowledge discussed in the previous section into practice. Certificate generation and other useful cryptographic tools are provided by the *OpenSSL* package.

```
# yum install openssl
```

Openssl's configuration file is located on '/etc/pki/tls/openssl.cnf'. All changes that we're going to perform are optional but recommended.

```
# vi /etc/pki/tls/openssl.cnf
dir = /etc/ssl          # Where everything is kept
countryName_default    = ES
stateOrProvinceName_default = Vizcaya
localityName_default   = Bilbao
organizationName_default = Universidad del Pais Vasco
organizationalUnitName = Laboratorio de planificacion \
de redes y servicios
```

Next we need to create the directories where certificates will be stored, as well as an index and a serial file. These files will be needed in the certification generation process.

```
# mkdir /etc/ssl
# cd /etc/ssl
# mkdir private
# mkdir newcerts
# touch index.txt
# echo '01' > serial
```

The first step is to create the Certificate Authority (CA).

```
# openssl req -new -x509 -extensions v3_ca -keyout \
private/cakey.pem -out cacert.pem -days 3650
```

Once the CA is created, we can start making certificate requests and signing them. In this case we're going to issue certificates for two servers and a client.

```
# openssl req -new -nodes -keyout spserver_key.pem \
-out spserver_req.pem -days 730
# openssl ca -policy policy_anything -out spserver_cert.pem \
-infiles /etc/ssl/spserver_req.pem
```

```
# openssl req -new -nodes -keyout npserver_key.pem -out \
npserver_req.pem -days 730
# openssl ca -policy policy_anything -out \
npserver_cert.pem -infiles /etc/ssl/npserver_req.pem
```

```
# openssl req -new -keyout client_key.pem -out \
client_req.pem -days 730
# openssl ca -policy policy_anything -out client_cert.pem \
-infiles /etc/ssl/client_req.pem
```

After completing these steps we've got 4 certificates (figure 1).

- CA certificate and the public/private key pair.
- Server 1, server 2 and client certificates, all signed by the CA, and the corresponding key pairs.

Next we can add the CA to the trusted CA's list. From now on all the certificates signed by this CA will be valid for us.

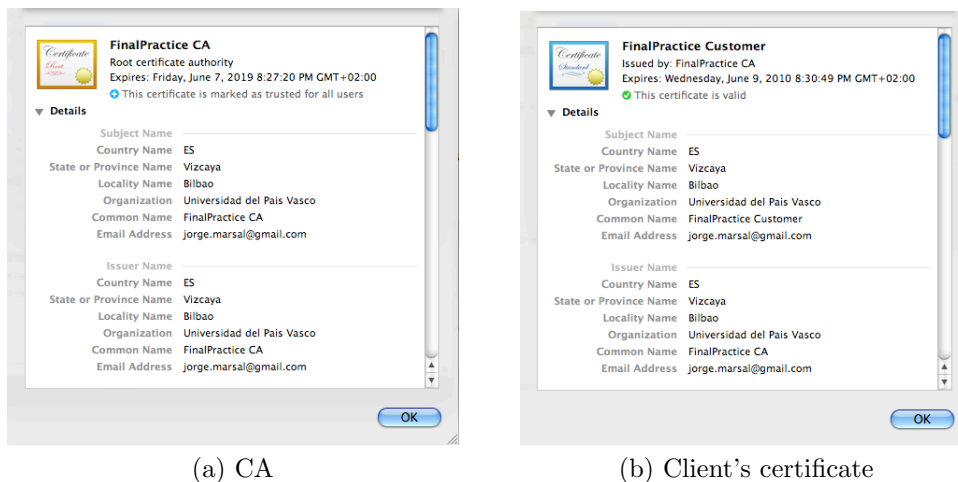


Figure 1: CA and client certs.

Figure 2 depicts the public key and the CA's signature on the client's certificate.

```
Public Key 128 bytes : B1 6C D9 4E CC 75 A2 97 A2 87 84 33 35 DC
3C 71 A0 C5 0A 9C C1 24 4F 97 FC 70 45 E0 4C 45 97 00
F6 9B 43 46 49 48 D8 20 B3 C8 11 F2 22 D6 4A 79 68 D2
02 B7 6A F9 1A 32 BD C7 B7 53 24 46 7B 6F 9E C5 DD 76
50 06 F9 42 0C ED 36 4E 8F A3 92 98 C4 16 90 E0 C4 DB
55 78 81 93 D8 E1 7D 2A 97 80 E0 4D 83 82 70 81 5A A5
82 36 C5 F0 7C 1C 1C 66 5A B1 A4 0D 94 A9 E4 39 4F 78
74 42 62 EB 3A 79

Exponent 65537
Key Size 1024 bits
Key Usage Any

Signature 128 bytes : 1E 7B E9 BF 5F A1 71 86 A1 CE FE 3F 5F 18 AF
C6 E6 B6 48 D4 D5 8E DF 46 E2 61 E1 A3 84 F4 FC ED 49
A4 D2 78 8E 75 67 41 43 77 8B 62 6C 22 EB 3E AD 95 1A
80 B2 20 A7 BD A8 47 74 1A E2 2A 9D 81 4C 98 FA 49 25
91 13 AC 28 DF 05 26 58 0E 5B 16 79 11 D1 3B 30 9E 60
3F E9 6D B6 01 03 B9 F0 D9 8E EF 2C 59 E6 D3 4F A2 B5
D0 E6 61 BF 9B A1 6E 82 57 C0 7A C8 75 B3 7F F7 F7 73
C7 C7 12 ED 4F
```

Figure 2: Public key and CA's sign in the client's cert.

Now that we've got the certificates we can use them for many purposes. For instance we can set up certificate-based 802.1X authentication on our network, enable SSL on a web server, etc.

References

- [1] The Public Key Infrastructure Approach to Security, *Oracle*, <http://www.oracle.com/>
- [2] Introduction to Digital Certificates, *Verisign*, <http://www.verisign.com/>