



## 06- Implementing Public Key Infrastructure

**Ahmed Sultan**

Senior Technical Instructor  
[ahmedsultan.me/about](https://ahmedsultan.me/about)

# Outlines

6.1- Implement Certificates and Certificate Authorities

6.2- Implement PKI Management

## Labs

Lab 6: Managing the Lifecycle of a Certificate

Lab 7: Managing Certificates with OpenSSL

## **6.1- Implement Certificates and Certificate Authorities**

### 6.2- Implement PKI Management

# PUBLIC AND PRIVATE KEY USAGE

- **Public key cryptography** solves the problem of distributing encryption keys when you want to communicate securely with others or authenticate a message that you send to others.
  - ✓ When you want others to send you **confidential** messages, you give them your **public key** to use to encrypt the message, The message can then only be decrypted by your **private key**, which you keep known only to yourself.
  - ✓ When you want to **authenticate** yourself to others, you create a signature and sign it by encrypting the signature with your **private key**, You give others your **public key** to use to decrypt the signature, As only you know the private key, everyone can be assured that only you could have created the signature.

# PUBLIC AND PRIVATE KEY USAGE (cont.)

- The basic problem with public key cryptography is that you may not really know with whom you are communicating.
- The system is vulnerable to man-in-the-middle attacks.
- This problem is particularly evident with e-commerce.
- How can you be sure that a shopping site or banking service is really maintained by whom it claims?
- The fact that the site is distributing public keys to secure communications is no guarantee of actual identity.

# PUBLIC AND PRIVATE KEY USAGE (cont.)

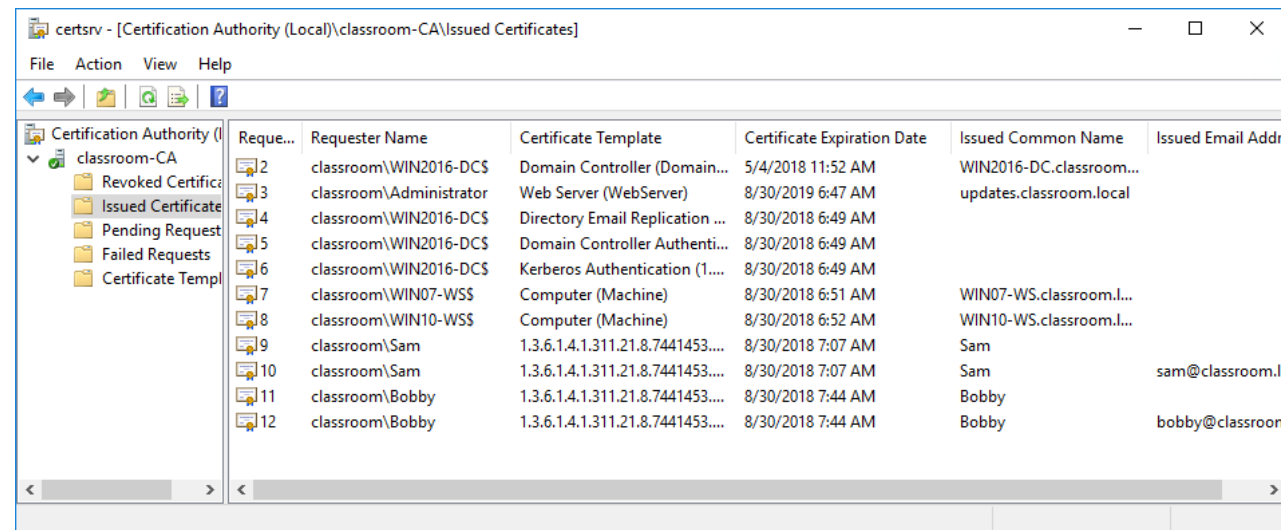
- How do you know that you are corresponding directly with the site using its certificate? How can you be sure there isn't a man-in-the-middle intercepting and modifying what you think the legitimate server is sending you? various models.
- **Public key infrastructure (PKI)** aims to prove that the owners of public keys are who they say they are.
- Under **PKI**, anyone issuing public keys should obtain a **digital certificate**.
- The validity of the certificate is guaranteed by a **certificate authority (CA)**.

# CERTIFICATE AUTHORITIES

- The **Certificate Authority (CA)** is the entity responsible for issuing and guaranteeing certificates.
- **Private CAs** can be set up within an organization for internal communications.
- Most network operating systems, including **Windows Server**, have certificate services.
- For public or business-to-business communications, however, the CA must be trusted by each party.
- Third-party CA services include **IdenTrust, Digicert, Sectigo/Comodo, GoDaddy, and GlobalSign**.

# CERTIFICATE AUTHORITIES (cont.)

- The functions of a CA are as follows:
  - ✓ Provide a range of certificate services useful to the community of users serviced by the CA.
  - ✓ Ensure the validity of certificates and the identity of those applying for them (registration).
  - ✓ Establish trust in the CA by users and government and regulatory authorities and enterprises, such as financial institutions.



The screenshot shows the 'certsrv' console window for the local Certification Authority 'classroom-CA'. The left pane shows the tree structure with 'Issued Certificate' selected. The right pane displays a table of issued certificates.

Reque...	Requester Name	Certificate Template	Certificate Expiration Date	Issued Common Name	Issued Email Address
2	classroom\WIN2016-DC\$	Domain Controller (Domain...	5/4/2018 11:52 AM	WIN2016-DC.classroom...	
3	classroom\Administrator	Web Server (WebServer)	8/30/2019 6:47 AM	updates.classroom.local	
4	classroom\WIN2016-DC\$	Directory Email Replication ...	8/30/2018 6:49 AM		
5	classroom\WIN2016-DC\$	Domain Controller Authent...	8/30/2018 6:49 AM		
6	classroom\WIN2016-DC\$	Kerberos Authentication (1...	8/30/2018 6:49 AM		
7	classroom\WIN07-WSS	Computer (Machine)	8/30/2018 6:51 AM	WIN07-WS.classroom.l...	
8	classroom\WIN10-WSS	Computer (Machine)	8/30/2018 6:52 AM	WIN10-WS.classroom.l...	
9	classroom\Sam	1.3.6.1.4.1.311.21.8.7441453....	8/30/2018 7:07 AM	Sam	
10	classroom\Sam	1.3.6.1.4.1.311.21.8.7441453....	8/30/2018 7:07 AM	Sam	sam@classroom.lc
11	classroom\Bobby	1.3.6.1.4.1.311.21.8.7441453....	8/30/2018 7:44 AM	Bobby	
12	classroom\Bobby	1.3.6.1.4.1.311.21.8.7441453....	8/30/2018 7:44 AM	Bobby	bobby@classroom



# PKI TRUST MODELS

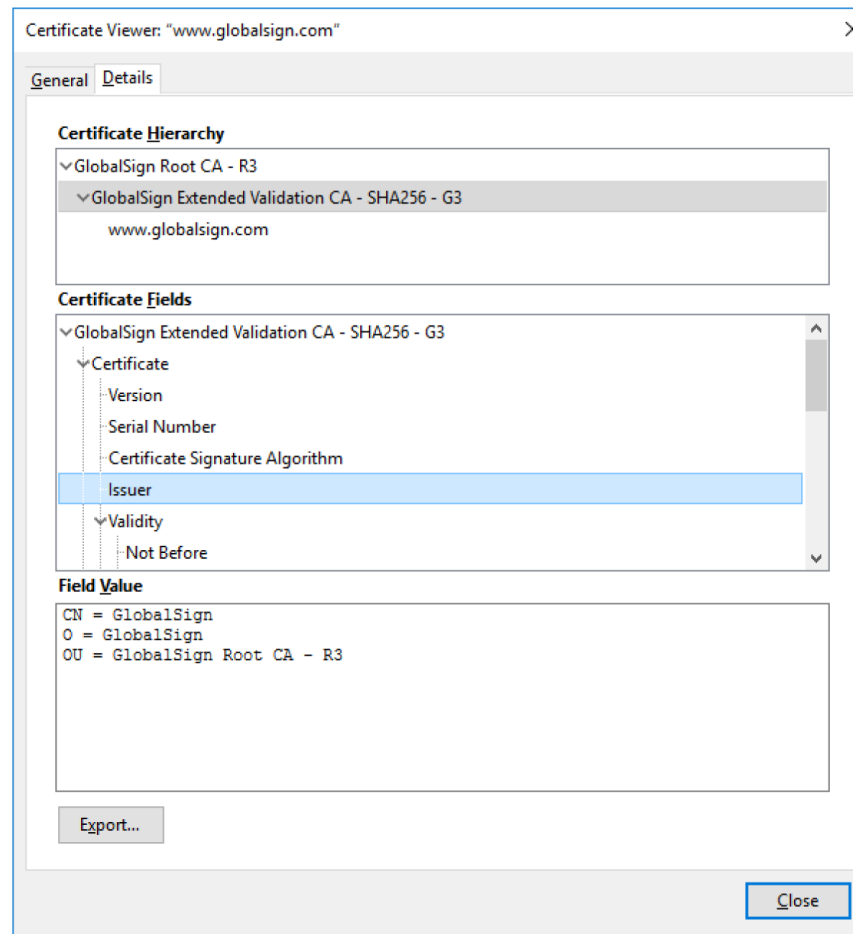
- The trust model is a critical PKI concept, and shows how users and different CAs are able to trust one another.
- **Single CA**
  - ✓ In this simple model, a single CA issues certificates to users; users trust certificates issued by that CA and no other.
  - ✓ The problem with this approach is that the single CA server is very exposed.
  - ✓ If it is compromised, the whole PKI collapses.

# PKI TRUST MODELS (cont.)

## ■ Hierarchical (Intermediate CA)

- ✓ In the hierarchical model, a single CA (called the root) issues certificates to several intermediate CAs.
- ✓ The intermediate CAs issue certificates to subjects (leaf or end entities).
- ✓ This model has the advantage that different intermediate CAs can be set up with different certificate policies, enabling users to perceive clearly what a particular certificate is designed for.
- ✓ Each leaf certificate can be traced back to the root CA along the certification path.
- ✓ This is also referred to as **certificate chaining**, or a **chain of trust**.
- ✓ The root's certificate is self-signed.
- ✓ In the hierarchical model, the root is still a single point of failure.
- ✓ If the root is damaged or compromised, the whole structure collapses.

# PKI TRUST MODELS (cont.)



# PKI TRUST MODELS (cont.)

- **Online vs. Offline CAs**

- ✓ An online CA is one that is available to accept and process certificate signing requests, publish certificate revocation lists, and perform other certificate management tasks.
- ✓ Because of the high risk posed by compromising the root CA, a secure configuration involves making the root an offline CA.
- ✓ This means that it is disconnected from any network and usually kept in a powered-down state.
- ✓ The root CA will need to be brought online to add or update intermediate CAs.

# REGISTRATION AUTHORITIES AND CSRS

- Registration is the process by which end users create an account with the CA and become authorized to request certificates.
- The exact processes by which users are authorized and their identity proven are determined by the CA implementation.
- **For Example:** in a Windows Active Directory network, users and devices can often auto-enroll with the CA just by authenticating to Active Directory.
- Commercial CAs might perform a range of tests to ensure that a subject is who he or she claims to be.
- It is in the CA's interest to ensure that it only issues certificates to legitimate users, or its reputation will suffer.

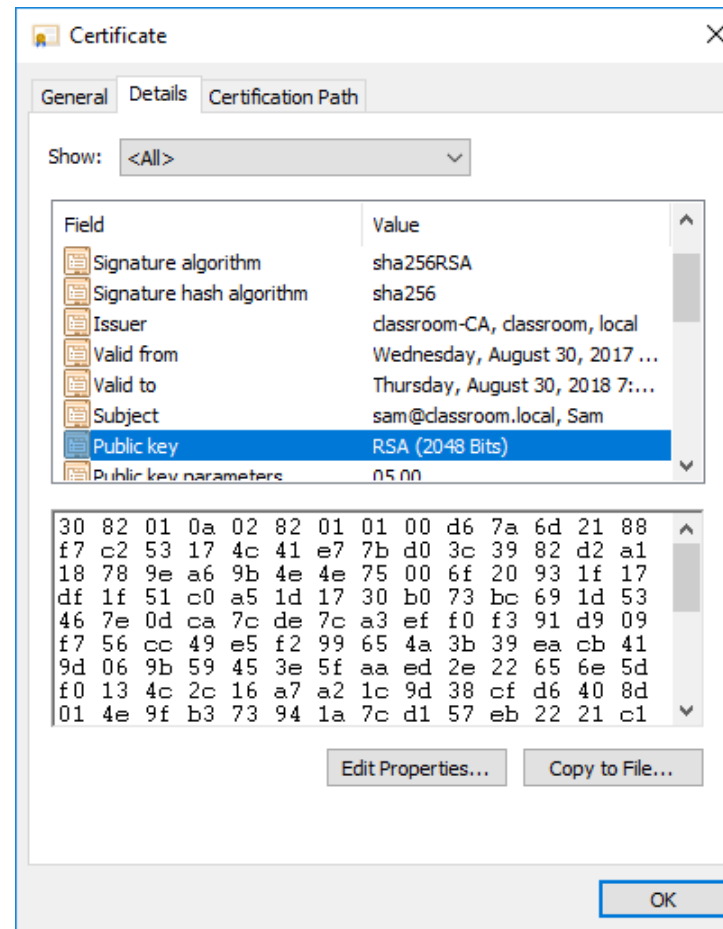
# REGISTRATION AUTHORITIES AND CSRS (cont.)

- When a subject wants to obtain a certificate, it completes a **certificate signing request (CSR)** and submits it to the CA.
- The **CSR** is a Base64 ASCII file containing the information that the subject wants to use in the certificate, including its public key.
- The CA reviews the certificate and checks that the information is valid.
- For a web server, this may simply mean verifying that the subject name and fully qualified domain name (FQDN) are identical, and verifying that the CSR was initiated by the person administratively responsible for the domain, as identified in the domain's WHOIS records.
- If the request is accepted, the CA signs the certificate and sends it to the subject.

# DIGITAL CERTIFICATES

- A **Digital Certificate** is essentially a wrapper for a subject's public key.
- As well as the public key, it contains information about the subject and the certificate's issuer or guarantor.
- The certificate is digitally signed to prove that it was issued to the subject by a particular CA.
- The subject could be a human user (for certificates allowing the signing of messages, for instance) or a computer server (for a web server hosting confidential transactions, for instance).

# DIGITAL CERTIFICATES (cont.)





# DIGITAL CERTIFICATES (cont.)

- Digital certificates are based on the **X.509** standard approved by the **International Telecommunications Union and standardized by the Internet Engineering Taskforce** ([tools.ietf.org/html/rfc5280](https://tools.ietf.org/html/rfc5280)).
- The Public Key Infrastructure (PKIX) working group manages the development of these standards.
- **RSA** also created a set of standards, referred to as Public Key Cryptography Standards (PKCS), to promote the use of public key infrastructure.

# CERTIFICATE ATTRIBUTES

- The **X.509** standard defines the fields or attributes that must be present in the certificate.

Field	Usage
Serial number	A number uniquely identifying the certificate within the domain of its CA.
Signature algorithm	The algorithm used by the CA to sign the certificate.
Issuer	The name of the CA.
Valid from/to	Date and time during which the certificate is valid.
Subject	The name of the certificate holder, expressed as a distinguished name (DN). Within this, the common name (CN) part should usually match either the fully qualified domain name (FQDN) of the server or a user email address.
Public key	Public key and algorithm used by the certificate holder.
Extensions	V3 certificates can be defined with extended attributes, such as friendly subject or issuer names, contact email addresses, and intended key usage.
Subject alternative name (SAN)	This extension field is the preferred mechanism to identify the DNS name or names by which a host is identified.

# Lab

## Lab 6: Managing the Lifecycle of a Certificate

6.1- Implement Certificates and Certificate Authorities

## **6.2- Implement PKI Management**

# CERTIFICATE AND KEY MANAGEMENT

- Key management refers to operational considerations for the various stages in a key's life cycle.
- A key's life cycle may involve the following stages:
  - ✓ **Key generation**—creating a secure key pair of the required strength, using the chosen cipher.
  - ✓ **Certificate generation**—to identify the public part of a key pair as belonging to a subject (user or computer), the subject submits it for signing by the CA as a digital certificate with the appropriate key usage. At this point, it is critical to verify the identity of the subject requesting the certificate and only issue it if the subject passes identity checks.
  - ✓ **Storage**—the user must take steps to store the private key securely, ensuring that unauthorized access and use is prevented. It is also important to ensure that the private key is not lost or damaged.

# CERTIFICATE EXPIRATION

- Certificates are issued with a **limited duration**, as set by the CA policy for the certificate type.
- Root certificates might have long expiration dates (**10+ years**), whereas web server and user certificates might be issued for **1 year only**.
- Typically, a certificate is renewed before it expires.
- Where a user is in possession of a valid certificate, less administration is required (in terms of checking identity) than with a request for a new certificate.
- When you are renewing a certificate, it is possible to use the existing key (referred to specifically as key renewal) or generate a new key (the certificate is rekeyed).

# CERTIFICATE REVOCATION LISTS

- A certificate may be revoked or suspended:
  - ✓ A revoked certificate is no longer valid and cannot be "un-revoked" or reinstated.
  - ✓ A suspended certificate can be re-enabled.
- A certificate may be revoked or suspended by the owner or by the CA for many reasons.
- **For Example:**
  - ✓ The certificate or its private key may have been compromised.
  - ✓ The business could have closed.
  - ✓ A user could have left the company.
  - ✓ A domain name could have been changed.
  - ✓ The certificate could have been misused in some way, and so on.

# OPENSSL

- In a [Windows environment](#), certificate infrastructure is installed and managed as **Active Directory Certificate Services**.
- There is a ***certutil*** tool for command line management, or you can use PowerShell.
- For [Linux](#), CA services are typically implemented using the **OpenSSL suite** ([openssl.org](https://openssl.org)).



# Lab

## Lab 7: Managing Certificates with OpenSSL