HERRAMIENTAS

OpenSSL es un paquete criptográfico disponible en Linux, por ejemplo en Ubuntu. Preste atención a los cambios que se van produciendo en el directorio en el que se ejecutan los comandos indicados.

OPENSSL. Disponible en el aula (Linux), información para Windows: https://wiki.openssl.org/index.php/Binaries

- En Windows, para poder ejecutarlo desde cualquier ruta del sistema debe incluir la carpeta bin de OpenSSL dentro de la variable de entorno PATH. Utilice el comando: set PATH=%PATH%;"PATH DONDE INSTALE OPENSSL"/bin

INTRODUCCIÓN

La práctica se estructura en 4 partes: 1) Creación de una PKI, 2) Firma digital, 3) Obtención de un certificado digital a través de una web, 4) Obtención del certificado digital de la Fábrica Nacional de Moneda y Timbre (Opcional)

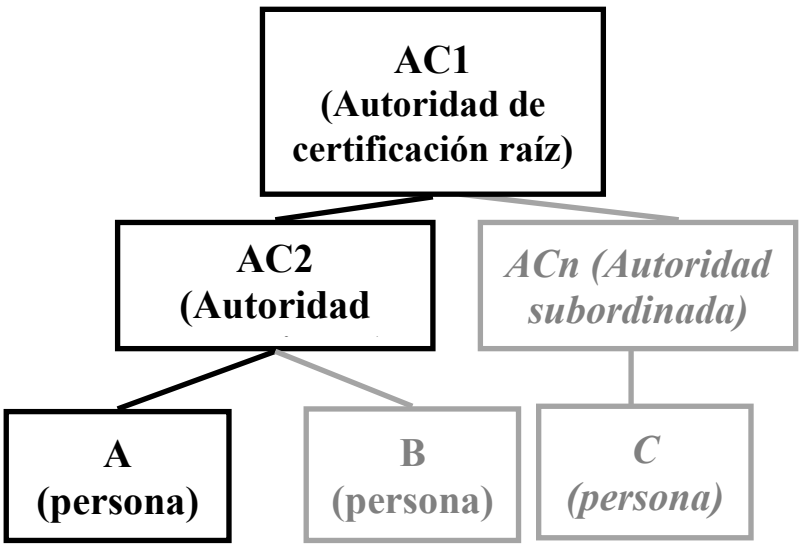
El objetivo de esta práctica es comprender los **fundamentos** sobre los que se basan las **infraestructuras de clave pública**. Particularmente, los objetivos concretos son los siguientes:

1. Comprender los pasos necesarios para que una Autoridad emita un certificado.
2. Entender qué papel juegan los certificados en la firma y verificación de documentos.

Para alcanzar estos objetivos, en esta práctica cada grupo de alumnos se convierte en una AUTORIDAD DE CERTIFICACIÓN RAÍZ (como puede ser en el mundo real, la Fábrica Nacional de Moneda y Timbre). Dicha Autoridad (AC1), por cuestiones organizativas (por ejemplo, para tener una delegación en cada comunidad autónoma) tiene varias AUTORIDADES DE CERTIFICACIÓN

SUBORDINADAS (AC2,…, ACn), las cuales se dedican a emitir certificados de clave pública a las personas (A, B, C).

El conjunto de todas estas Autoridades conforma una INFRAESTRUCTURA DE CLAVE PÚBLICA (en inglés, PKI).



Para limitar la carga de trabajo, en esta práctica sólo se gestionará la autoridad raíz (AC1), una única autoridad subordinada (AC2) y una persona (A).

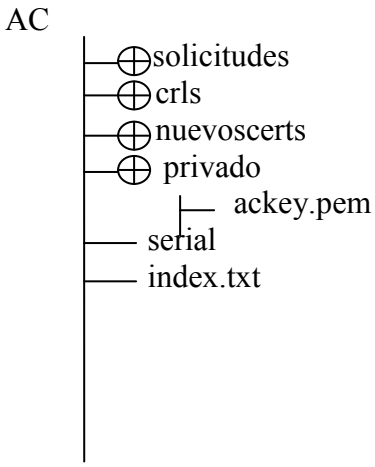
Para organizar el desarrollo de la práctica, cree tres carpetas, una cada para entidad: AC1, AC2 y A.

```
# practica> mkdir AC1 AC2 A
```

Para emitir los certificados, las Autoridades utilizan una POLÍTICA de certificación. Copie los ficheros que contienen estas políticas openssl_AC1.cnf y openssl_AC2.cnf (disponibles en Aula Global) en el directorio AC1 y AC2 respectivamente. Analice los ficheros openssl_ACx.cnf dados, comparándolos entre sí y con el fichero de configuración por defecto, el cual se encuentra en /etc/ssl/*openssl*.cnf.

Antes de comenzar la práctica, modifique los ficheros de políticas de tal forma que el nombre de sus AC sea AC1-XXXXX y AC2-XXXXX, donde XXXXX son los cinco últimos dígitos de su identificador de alumno en la Universidad.

La estructura de los directorios y archivos quedará en un principio como sigue:



Configuración de AC1 (Autoridad de Certificación raíz)

Los comandos de OpenSSL necesarios para la realización de la práctica son:

- ca: permite crear y gestionar una Autoridad de Certificación basada en el modelo de confianza jerárquico.
- req: permite crear y gestionar peticiones de emisión de certificados X.509.
- x509: permite gestionar certificados X.509.
- verify: permite verificar certificados X.509.

1. Genere la estructura de directorios necesaria para AC1 e inicialice los ficheros serial e index.txt.

   # AC1> mkdir solicitudes crls nuevoscerts privado

   # AC1> echo '01' > serial

   # AC1> touch index.txt

2. Genere un par de claves RSA junto con el certificado autofirmado por AC1. Estudie los cambios producidos en el directorio AC1.

   # AC1> openssl req -x509 -newkey rsa:2048 -days 360 -out ac1cert.pem -outform PEM -config openssl_AC1.cnf

   Se pide un passphrase para crear la clave privada de AC1, que habrá que recordar cuando queramos utilizarla.

   # AC1> openssl x509 -in ac1cert.pem -text -noout

Configuración de AC2 (Autoridad de Certificación subordinada)

3. Genere la estructura de directorios necesaria para AC2 e inicialice los ficheros serial e index.txt.

   # AC2> mkdir solicitudes crls nuevoscerts privado

   # AC2> echo '01' > serial

   # AC2> touch index.txt

4. Genere un par de claves de RSA junto con una solicitud de emisión de certificado y "envíesela" a AC1. Estudie los cambios producidos en el directorio AC2.

AC2> openssl req -newkey rsa:2048 -days 360 -out ac2req.pem -outform PEM -config openssl_AC2.cnf

Al igual que antes en AC1, se pide un passphrase que habrá que recordar cuando se utilice la clave privada de AC2

AC2> openssl req -in ac2req.pem -text -noout

AC2> cp ac2req.pem ../AC1/solicitudes

Generación del certificado de AC2 por AC1

5. Verifique la solicitud de emisión de certificado "enviada" por AC2.

AC1> openssl req -in ./solicitudes/ac2req.pem -text -noout

6. Genere el certificado de AC2 y "envíeselo" (en el proceso, cambie el nombre del nuevo certificado - actualmente 01.pem - a ac2cert.pem, ya que AC2 tiene configurado en su fichero de configuración este último nombre). Estudie los cambios producidos en el directorio AC1.

   # AC1> openssl ca -in ./solicitudes/ac2req.pem -notext -extensions v3_subca -config openssl_AC1.cnf

   AC1 utiliza su clave privada para crear el certificado de AC2, por lo que se pide el passphrase de AC1.

   # AC1> cp ./nuevoscerts/01.pem ../AC2/ac2cert.pem

Generación de las claves de A y su solicitud de emisión de certificado para AC2

7. Para la entidad A, genere un par de claves de RSA junto con una solicitud de emisión de certificado y "envíeselas" a AC2 (en el proceso de generación de solicitudes de emisión de certificado, rellene todos los campos que se le soliciten e indique que el país es "ES", que la provincia es "MADRID", que la organización es "UC3M", que el nombre común es XXXXX (según lo indicado anteriormente), y que su email es su dirección de correo de estudiante). Estudie los cambios producidos en el directorio A.

   # A> openssl req -newkey rsa:1024 -days 360 -sha1 -keyout Akey.pem -out Areq.pem

Al igual que antes, se pide un passphrase que habrá que recordar cuando se quiera utilizar la clave privada de A.

```
# A> openssl req -in Areq.pem -text -noout
# A> cp Areq.pem ../AC2/solicitudes
```

Generación del certificado de A por AC2

8. Compruebe la solicitud de emisión de certificados "enviada" por A.

   ```
   # AC2> openssl req -in ./solicitudes/Areq.pem -text -noout
   ```

9. Genere el certificado de A y "envíeselo" de vuelta (en el proceso, cambie el nombre del nuevo certificado - actualmente 01.pem - a Acert.pem).

   ```
   # AC2> openssl ca -in ./solicitudes/Areq.pem -notext -config ./openssl_AC2.cnf
   ```

   AC2 utiliza su clave privada para crear el certificado de A, por lo que se pide el passphrase de AC2.

   ```
   # AC2> cp ./nuevoscerts/01.pem ../A/Acert.pem
   ```

10. Estudie los cambios producidos en el directorio AC2 y compruebe el certificado resultante:

    ```
    # A> openssl x509 -in Acert.pem -text -noout
    ```

Verificación del certificado de A

11. Obtenga una copia auténtica de los certificados de clave pública de AC1 y AC2 y verifique el certificado de A (para ello necesitara concatenar los certificados de AC1 y AC2 en un único fichero).

    ```
    # A> cp ../AC1/ac1cert.pem ./
    # A> cp ../AC2/ac2cert.pem ./
    # A> cat ac1cert.pem ac2cert.pem > certs.pem
    # A> openssl verify -CAfile certs.pem Acert.pem
    ```

    Si todo es correcto, marcará un OK al ejecutar el último comando