SEGURIDAD Y PROTECC.DE SIST.INFORMÁTICOS (2018-2019)
Grado en Ingeniería Informática
Universidad de Granada

Práctica 4. Certificados digitales

Patricia Maldonado Mancilla

Índice

Índice de figuras

1. Cread una autoridad certificadora raiz. Mostrad los archivos creados y sus rutas, y los valores de las claves generadas.

Una **autoridad certificadora (CA)** es una entidad con capacidad para firmar certificados. Para crear una CA raíz es necesario crear el par criptográfico llave privada (cakey.pem) y certificado público (cacert.pem). Esta CA puede ser usada para firmar certificados de autoridades certificadores intermedias o certificados finales de usuarios o servidores.
Tenemos que crear una estructura de directorios adecuada y las llaves privada y pública, así como un fichero índice para los certificados generados por esta autoridad.

1. **Empezamos creando la estructura de directorios.**

```
patri@patri:~/Escritorio$ mkdir CA
patri@patri:~/Escritorio$ cd CA/
patri@patri:~/Escritorio/CA$ mkdir certs crl newcerts private
```

Figura 1.1: Escructura de direcctorios para CA raíz

- **CA:** directorio de trabajo para la autoridad certificante raíz.
- **certs:** directorio donde se ubicarán los certificados.
- **crl:** directorio donde se ubicará la lista de renovación de certificados.
- **newcerts:** directorio con los certificados con formato PEM.
- **private:** directorio donde se colocan las claves privadas.

2. **Creamos dos archivos que funcionan como bases de datos para Openssl, el fichero índice y el serial. El archivo serial almacena el numero del siguiente certificado firmado y el archivo índice donde se almacenará la base de datos de los certificados.**

```
patri@patri:~/Escritorio/CA$ touch serial
patri@patri:~/Escritorio/CA$ echo 1000 > serial
patri@patri:~/Escritorio/CA$ touch index.txt
```

Figura 1.2: Fichero índice y serial

3. **Para la configuración inicial de Openssl, tenemos que copiar el archivo de configuración openssl.cnf a nuestra carpeta CA.**

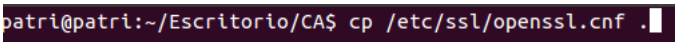

Figura 1.3: Copia archivo openssl.cnf a la carpeta CA

En este caso solo he cambiado la ruta de la CA raíz y la ruta donde se almacenan los certificados.

```
5 ####################################################################
6 [ ca ]
7 default_ca  = CA_default        # The default ca section
8
9 ####################################################################
0 [ CA_default ]
1
2 dir     = /home/patri/Escritorio/CA     # Where everything is kept
3 certs       = $dir/certs        # Where the issued certs are kept
4 crl_dir     = $dir/crl      # Where the issued crl are kept
5 database    = $dir/index.txt    # database index file.
6 #unique_subject = no            # Set to 'no' to allow creation of
7                     # several ctificates with same subject.
8 new_certs_dir   = $dir/newcerts     # default place for new certs.
9
0 certificate = $dir/certs/cacert.pem     # The CA certificate
1 serial      = $dir/serial       # The current serial number
2 crlnumber   = $dir/crlnumber    # the current crl number
3                     # must be commented out to leave a V1 CRL
4 crl     = $dir/crl.pem      # The current CRL
5 private_key = $dir/private/cakey.pem# The private key
6 RANDFILE    = $dir/private/.rand    # private random number file
```

Figura 1.4: Modificaciones openssl.cnf CA raíz

Se puede personalizar aún más para definir políticas para la creación y firmado de los certificados, o definir extensiones deseadas para nuevos certificados. Podemos configurar con más detalle el archivo de configuración de openssl, a parte de las rutas donde se almacenan los certificados y las claves privadas, podemos definiir la funcion hash que vamos a utlizar en la linea default_md = sha256, dentro de la sección req tamaño por defecto, default_bits = 2048, en la sección v3_req las extensiones para agregar a una solicitud de certificado...

4. **Creamos un par de claves RSA para la entidad certificadora CA raíz.** Lo realizamos mediante el comando **openssl genrsa** indicando que el algoritmo de cifrado **aes256**, la crearemos con un tamaño de 2048 bits y se guardará mediante el argumento **-out** con el nombre cakey.pem en la carpeta private.

```
patri@patri:~/Escritorio/CA$ openssl genrsa -aes256 -out private/cakey.pem 2048
Generating RSA private key, 2048 bit long modulus
.....................................+++
..............+++
e is 65537 (0x10001)
Enter pass phrase for private/cakey.pem:
Verifying - Enter pass phrase for private/cakey.pem:
```

Figura 1.5: Creación del par de claves RSA para CA raíz

Podemos visualizarla mediante el comando:
openssl rsa -in private/cakey.pem -text -noout

```
Private-Key: (2048 bit)
modulus:
    00:d2:98:69:66:ad:99:34:1b:25:77:c4:bf:99:54:
    92:ca:2f:e5:15:37:1f:79:cd:7e:c0:8e:91:21:55:
    40:88:85:77:f4:50:90:ff:ae:25:7a:53:1c:55:02:
    de:d9:a2:7b:d9:ba:1c:b2:d0:ba:ab:64:e2:e8:92:
    56:e3:5a:6e:f4:2e:4b:68:24:e4:78:1b:16:fe:b3:
    7e:ce:84:5d:2b:a8:ab:dc:70:be:34:e4:e4:8a:d0:
    51:db:cd:72:74:c3:2c:ac:c8:05:b9:6f:4e:93:f5:
    59:c0:bf:3a:64:c7:52:95:6a:77:29:35:14:96:d3:
    f1:32:33:eb:0d:c3:eb:cd:88:87:cb:40:33:eb:71:
    4a:6e:b6:e1:0f:4c:7e:d2:fe:7c:86:42:4e:5d:22:
    b8:3b:66:59:ec:4b:4d:6e:88:53:08:52:c8:05:f1:
    47:07:b3:7c:8b:f7:d4:ee:07:9d:a6:29:c8:14:8d:
    8d:e3:bc:3d:b1:17:15:67:93:18:9d:44:d6:bc:f8:
    8a:20:7a:71:76:09:3c:eb:2f:fa:d5:5b:d7:ac:69:
    e9:aa:6c:ad:dc:23:aa:1d:a5:ad:ab:0d:dd:25:87:
    f8:76:47:38:ce:1e:6c:62:5f:b4:05:ff:0b:2d:a8:
    a4:af:57:d2:7e:c5:62:dd:b9:32:2a:8e:73:af:d7:
    67:0b
publicExponent: 65537 (0x10001)
privateExponent:
    38:e4:95:ec:f8:4f:20:45:89:2c:bf:09:67:f9:f2:
    22:0f:28:5b:3f:e6:0f:f8:d5:1a:5e:29:95:10:bb:
    fb:65:db:4f:34:88:c7:9a:1b:63:dd:9a:14:d9:4e:
    ad:0a:f9:5c:8b:ab:ff:1e:e6:9f:31:05:20:01:90:
    d9:a0:ef:c2:a5:d3:17:52:88:2a:6e:5f:1a:f6:10:
    f6:e7:c9:67:e7:0c:6e:e4:42:6e:75:bb:cc:48:95:
    24:c5:6e:fb:5e:52:ee:66:a7:2d:3e:85:c2:f1:10:
    8b:42:de:1a:a5:c2:88:bd:17:6e:19:79:f3:79:54:
    7a:d4:8d:d6:cd:2c:21:fd:17:dd:d4:a8:1c:89:5c:
    81:4f:3f:a0:1e:84:4b:ab:83:fd:d3:3c:fa:74:de:
    9b:5f:10:e9:4a:e9:d3:d1:20:64:d5:3a:16:aa:a0:
    a4:19:82:b9:23:63:01:bf:f0:54:7b:69:94:1e:5a:
    e4:23:d4:ff:3e:d5:55:f5:b2:59:88:21:1b:ab:b5:
    cc:62:cf:b9:95:46:43:a1:13:4e:0e:a4:73:b8:b0:
    34:88:d3:d8:02:21:1b:4f:03:4f:d1:19:88:db:62:
    96:bb:22:77:2d:df:ba:a8:45:2f:72:f3:b9:d4:c7:
    8e:8d:17:fa:55:3b:3a:a8:82:ec:7d:50:18:42:8f:
    61
```

Figura 1.6: Contenido par de claves RSA para CA raíz

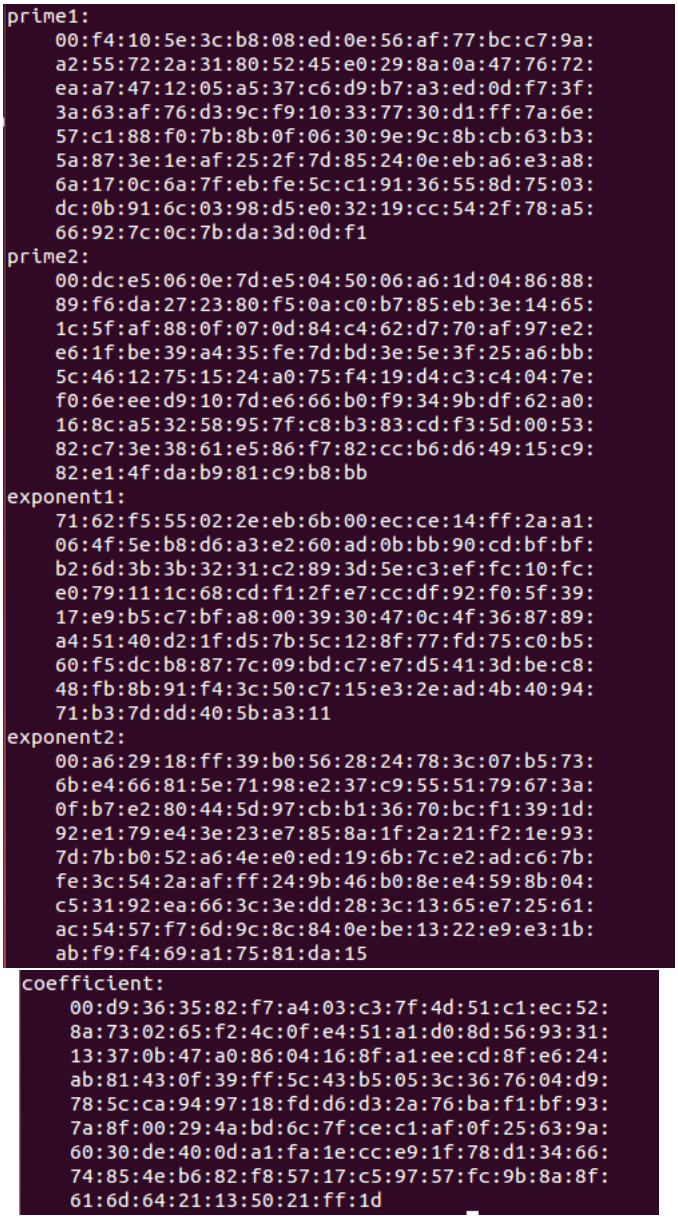

Figura 1.7: Contenido par de claves RSA para CA raíz

5. **Por último creamos el certificado autofirmado de la entidad certificadora CA raíz.**
   Utilizamos el siguiente comando:
   openssl req -x509 -new -config openssl.cnf -key private/cakey.pem -sha256 -out certs/cacert.pem.

   Con **openssl req** podemos crear solicitudes de certificado y generar certificados. En este caso vamos a crear un certificado autofirmado. Los argumentos que hemos utilizado en este comando son los siguientes:

   - **-x509:** esta opción es la que nos permite indicar que vamos a crear un certificado autofirmado y no una solicitud de certificado.
   - **-new:** crea una nueva solicitud de certificado.
   - **-sha256:** algoritmo de cifrado que usamos.
   - **-key:** especifica el archivo para leer la clave privada. La clave privada cakey.pem creada en el paso anterior que está almacenada en la carpeta private.
   - **-config:** permite especificar el archivo de configuración alternativo.
   - **-out:** para guardar nuestro certificado autofirmado, que lo hemos ubicado en la carpeta certs con el nombre cacert.pem.

```
patri@patri:~/Escritorio/CA$ openssl req -x509 -new -config openssl.cnf -key pri
vate/cakey.pem -sha256 -out certs/cacert.pem
Enter pass phrase for private/cakey.pem:
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:ES
State or Province Name (full name) [Some-State]:Granada
Locality Name (eg, city) []:Granada
Organization Name (eg, company) [Internet Widgits Pty Ltd]:SPSI
Organizational Unit Name (eg, section) []:SPSI-CAr
Common Name (e.g. server FQDN or YOUR name) []:SPSI-CAr
Email Address []:
```

Figura 1.8: Creación del certificado para CA raíz

Aparte de haberse creado el certificado cacert.pem, a la vez se guarda una copia del certificado, los archivos index y serial se actualizan, en index.txt se almacena la información del certificado creado con el número de serial asociado,y además se incrementa el valor en el archivo de serial.

Visualizamos el contenido del certificado creado mediante el siguiente comando:

```
patri@patri:~/Escritorio/CA$ openssl x509 -noout -text -in certs/cacert.pem
```

Figura 1.9: Comando para visualizar el certificado de CA raíz

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 9985893768057031009 (0x8a95057979d38961)
    Signature Algorithm: sha256WithRSAEncryption
        Issuer: C=ES, ST=Granada, L=Granada, O=SPSI, OU=SPSI-CAr, CN=SPSI-CAr
        Validity
            Not Before: Nov 23 09:19:03 2018 GMT
            Not After : Dec 23 09:19:03 2018 GMT
        Subject: C=ES, ST=Granada, L=Granada, O=SPSI, OU=SPSI-CAr, CN=SPSI-CAr
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                Modulus:
                    00:d2:98:69:66:ad:99:34:1b:25:77:c4:bf:99:54:
                    92:ca:2f:e5:15:37:1f:79:cd:7e:c0:8e:91:21:55:
                    40:88:85:77:f4:50:90:ff:ae:25:7a:53:1c:55:02:
                    de:d9:a2:7b:d9:ba:1c:b2:d0:ba:ab:64:e2:e8:92:
                    56:e3:5a:6e:f4:2e:4b:68:24:e4:78:1b:16:fe:b3:
                    7e:ce:84:5d:2b:a8:ab:dc:70:be:34:e4:e4:8a:d0:
                    51:db:cd:72:74:c3:2c:ac:c8:05:b9:6f:4e:93:f5:
                    59:c0:bf:3a:64:c7:52:95:6a:77:29:35:14:96:d3:
                    f1:32:33:eb:0d:c3:eb:cd:88:87:cb:40:33:eb:71:
                    4a:6e:b6:e1:0f:4c:7e:d2:fe:7c:86:42:4e:5d:22:
                    b8:3b:66:59:ec:4b:4d:6e:88:53:08:52:c8:05:f1:
                    47:07:b3:7c:8b:f7:d4:ee:07:9d:a6:29:c8:14:8d:
                    8d:e3:bc:3d:b1:17:15:67:93:18:9d:44:d6:bc:f8:
                    8a:20:7a:71:76:09:3c:eb:2f:fa:d5:5b:d7:ac:69:
                    e9:aa:6c:ad:dc:23:aa:1d:a5:ad:ab:0d:dd:25:87:
                    f8:76:47:38:ce:1e:6c:62:5f:b4:05:ff:0b:2d:a8:
                    a4:af:57:d2:7e:c5:62:dd:b9:32:2a:8e:73:af:d7:
                    67:0b
                Exponent: 65537 (0x10001)
        X509v3 extensions:
```

Figura 1.10: Contenido certificado de CA raíz

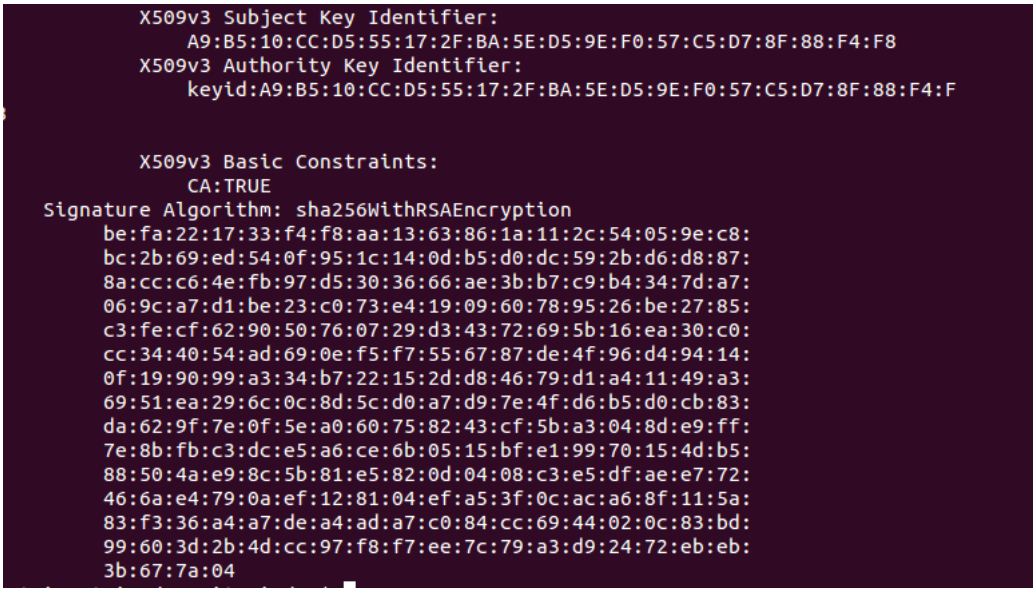

Figura 1.11: Contenido certificado de CA raíz

2. Cread una autoridad certificadora subordinada a la anterior. Mostrad los archivos creados y sus rutas, y los valores de las claves generadas.

En este apartado vamos a crear una **autoridad certificadora subordinada** que guardaremos en el directorio SUBCA. Seguiremos los mismos pasos que hemos seguido para realizar la creación de la CA raíz (estructura de directorios, creación de archivo índice y serial, generación de par de claves RSA), diferenciándose de que en este caso no creamos un certificado autofirmado, si no que crearemos una solicitud de certificado, que posteriormente firmará la CA raíz creando el certificado.

1. **Empezamos creando la estructura de directorios.**

```
patri@patri:~/Escritorio$ mkdir SUBCA
patri@patri:~/Escritorio$ cd SUBCA/
patri@patri:~/Escritorio/SUBCA$
patri@patri:~/Escritorio/SUBCA$ mkdir certs crl newcerts private
```

Figura 2.1: Escructura de direcctorios para SUBCA

2. **Creamos dos archivos que funcionan como bases de datos para Openssl, el fichero índice y el serial.**

```
patri@patri:~/Escritorio/SUBCA$ touch serial
patri@patri:~/Escritorio/SUBCA$ echo 1000 > serial
patri@patri:~/Escritorio/SUBCA$ touch index.txt
```

Figura 2.2: Fichero índice y serial

3. **Al igual que para la CA raíz, copiaremos el archivo de Openssl.** Además en este caso también he editado solamente la ruta donde se ubica nuestra subordinada y la ruta donde se almacenan los certificado.

```
patri@patri:~/Escritorio/SUBCA$ cp /etc/ssl/openssl.cnf .
```

Figura 2.3: Copia archivo openssl.cnf a la carpeta CA

```
35 ####################################################################
36 [ ca ]
37 default_ca  = CA_default        # The default ca section
38
39 ####################################################################
40 [ CA_default ]
41
42 dir     = /home/patri/Escritorio/SUBCA      # Where everything is kept
43 certs       = $dir/certs        # Where the issued certs are kept
44 crl_dir     = $dir/crl      # Where the issued crl are kept
45 database    = $dir/index.txt    # database index file.
46 #unique_subject = no            # Set to 'no' to allow creation of
47                     # several ctificates with same subject.
48 new_certs_dir   = $dir/newcerts     # default place for new certs.
49
50 certificate = $dir/certs/cacert.pem     # The CA certificate
51 serial      = $dir/serial       # The current serial number
52 crlnumber   = $dir/crlnumber    # the current crl number
53                     # must be commented out to leave a V1 CRL
54 crl     = $dir/crl.pem      # The current CRL
55 private_key = $dir/private/cakey.pem# The private key
56 RANDFILE    = $dir/private/.rand    # private random number file
57
58 x509_extensions = usr_cert      # The extentions to add to the cert
```

Figura 2.4: Modificaciones openssl.cnf CA subordinada

4. **Creamos el par de claves para la autoridad certificadora subordinada.** Esto lo hicimos anteriormente para la CA raíz, utilizamos el mismo comando.

```
patri@patri:~/Escritorio/SUBCA$ openssl genrsa -aes256 -out private/cakey.pem 20
48
Generating RSA private key, 2048 bit long modulus
.....................................................................+
++
....+++
e is 65537 (0x10001)
Enter pass phrase for private/cakey.pem:
Verifying - Enter pass phrase for private/cakey.pem:
```

Figura 2.5: Creación del par de claves para autoridad certificadora subordinada

Podemos visualizarla mediante el comando:
openssl rsa -in private/cakey.pem -text -noout

```
Private-Key: (2048 bit)
modulus:
    00:ba:4c:16:5b:c2:2d:aa:d9:12:bd:9c:28:0f:9e:
    9b:0c:35:72:0a:ab:dc:e4:85:ae:03:6b:26:a3:ce:
    82:a0:45:87:ce:0d:25:ba:9d:98:ee:76:07:17:57:
    6c:80:1f:1c:12:50:af:3a:b9:cc:f1:07:6f:68:17:
    e9:4e:6d:d0:66:a9:12:83:c9:1b:a3:d7:5b:9c:5e:
    87:99:94:b7:60:a6:85:7d:9a:f9:95:cf:3e:32:a9:
    4b:71:68:40:2c:5e:ab:6c:02:19:4f:04:c4:62:48:
    fa:85:1b:f8:4a:c8:3a:56:fe:ab:53:6f:17:fc:03:
    92:37:43:71:ae:ab:70:59:19:96:a8:e4:a0:a7:45:
    05:e4:b6:92:ad:5d:c7:bf:80:da:81:9b:58:fe:8f:
    ab:88:26:5f:42:18:bd:cc:63:0e:b1:e3:af:3a:1f:
    e4:01:ed:b4:fd:c0:cc:59:1e:43:45:c5:89:15:a1:
    11:90:e1:dd:d9:28:59:e7:98:15:33:da:05:df:13:
    ce:0d:b3:01:ae:8d:fa:31:82:31:38:29:3f:30:f7:
    06:59:5b:5b:69:0c:25:13:68:33:bc:b0:b1:b6:81:
    8d:20:18:6f:5c:e8:55:a8:f5:d9:00:b6:28:48:90:
    d1:18:81:5f:3d:99:39:5d:b5:14:23:b3:8d:c1:cc:
    7e:bd
publicExponent: 65537 (0x10001)
privateExponent:
    1c:90:0d:dd:83:a7:fc:c9:cb:55:c6:db:5d:89:b3:
    bb:eb:75:d0:ad:08:f1:00:8d:ed:78:40:34:5d:35:
    29:dd:7f:34:d8:4f:30:8d:79:25:16:73:7f:e0:a7:
    81:fd:43:10:8b:ce:51:f2:aa:b9:49:aa:8b:0b:1f:
    7a:26:dc:0c:e1:5f:35:55:60:27:7e:9d:4d:1c:31:
    2e:1a:df:02:b2:85:44:fc:bd:c7:7a:3e:cf:05:a5:
    cb:7d:70:59:5d:2a:6b:24:e7:3c:76:3e:a5:d3:0a:
    95:dd:f3:f3:83:3f:6e:97:1d:2d:6d:9a:ff:c8:8c:
    62:6c:b4:c1:53:07:a6:a1:42:35:92:ea:ba:8c:9b:
    7d:ba:08:d8:41:fa:88:15:f5:97:06:32:61:de:e9:
    94:f0:ca:35:8b:ae:43:90:cd:f0:08:6e:7f:e5:54:
    72:08:bd:2a:0f:5c:7a:a0:03:44:48:7c:05:99:16:
    04:09:c6:7e:85:15:02:bf:4f:b4:69:ad:66:1b:11:
    da:cd:36:3c:cd:a6:ed:8c:44:e0:e1:c8:6e:69:28:
    5e:45:a3:9c:b9:ab:ce:68:98:1b:2b:38:55:a8:c2:
    a1:31:39:e5:f3:76:c0:04:f5:aa:4b:74:dc:b1:2e:
    22:2d:96:8d:d2:0c:d2:d0:54:cf:9f:66:81:62:73:
    01
```

Figura 2.6: Contenido par de claves para CA subordinada

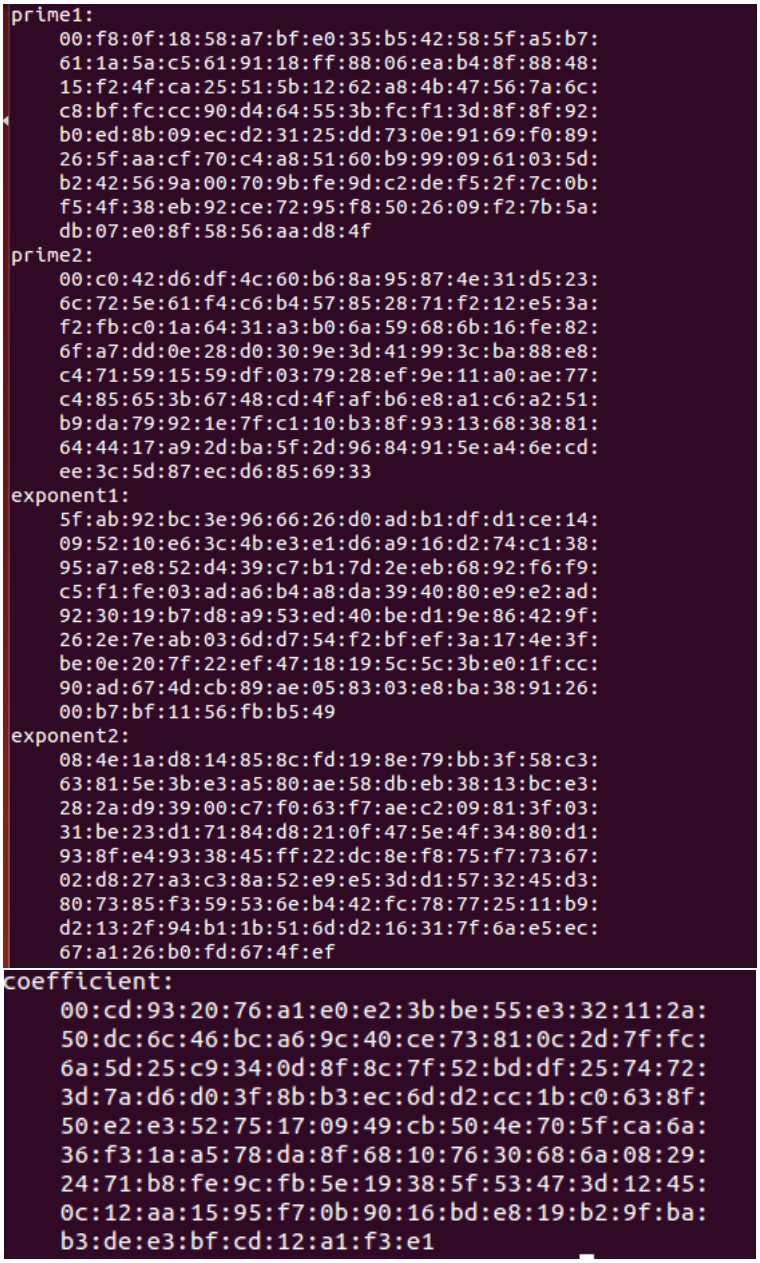

Figura 2.7: Contenido par de claves para CA subordinada

5. **Creamos una solicitud de certificado para la CA subordinada.**

   Mediante el siguiente comando realizamos la creación de la solicitud de certificado:

```
/SUBCA$ openssl req -config openssl.cnf -key private/cakey.pem -new -out solicitud_sub.pem
```

Figura 2.8: Comando solicitud CA subordinada

En este caso como queremos crear una solicitud de certificado no especificamos el argumento -x509, por lo que considerará que es una solicitud y no un certificado autofirmado. Además al comando openssl req, le pasamos el archivo que contiene el par de claves mediante el argumento **-key** cakey.pem situada en la carpeta private, -new para crear una nueva solicitud, además de especificarle el archivo de configuración mediante -config y lo guardamos con el nombre de solicitud_sub.pem

El contenido de la solicitud creada podemos visualizarlo mediante el siguiente comando:

```
patri@patri:~/Escritorio/SUBCA$ openssl req -verify -in solicitud_sub.pem -text -noout
```

Figura 2.9: Comando para visualizar solicitud CA subordinada

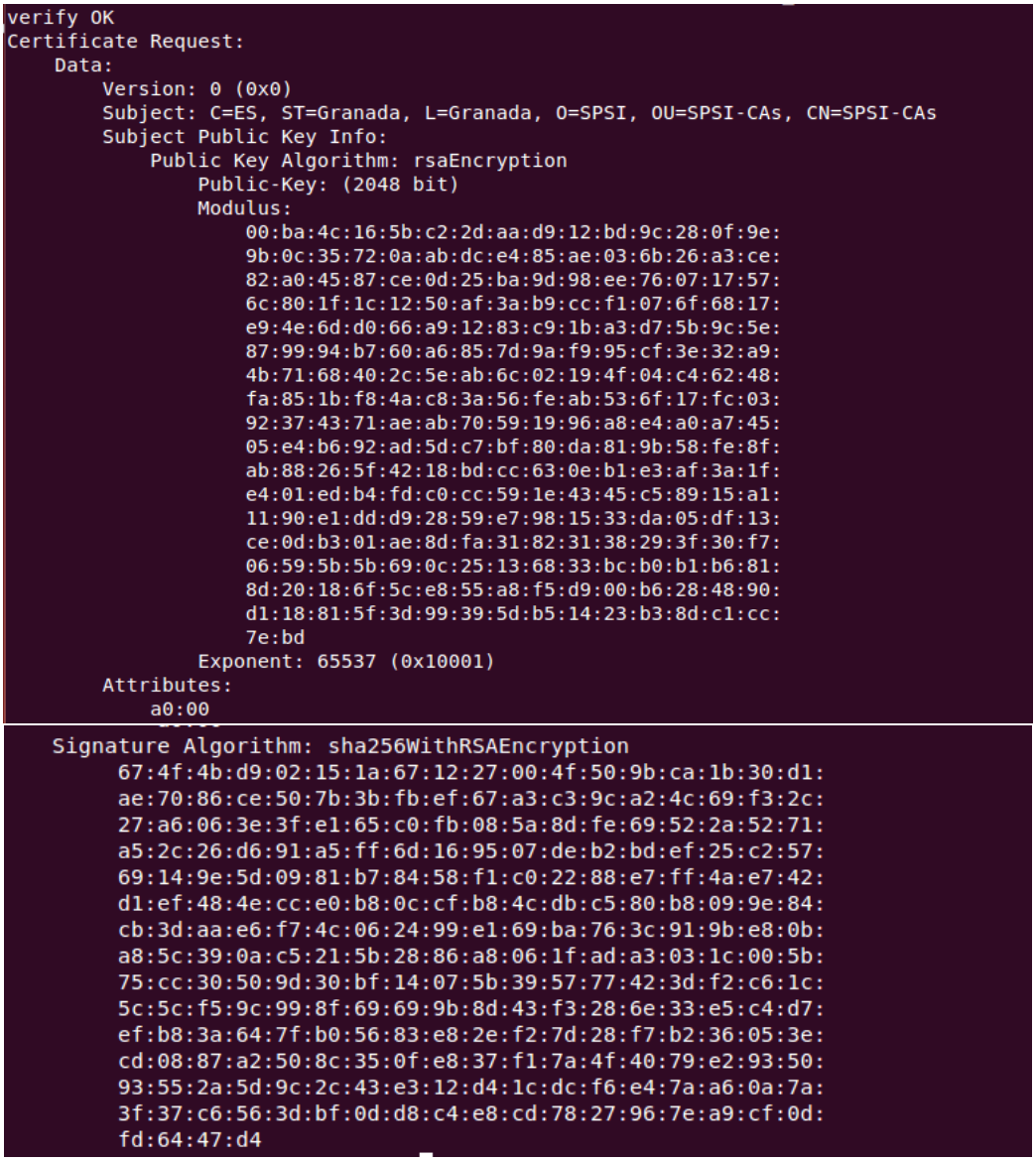

Figura 2.10: contenido de la solicitud CA subordinada

6. La CA raíz firma la solicitud de certificado de la CA subordinada.

Utilizamos el siguiente comando:

openssl ca -in solicitud_sub.pem -config openssl.cnf -extensions v3_a -out cacertsub.pem

Openssl ca se puede utilizar para firmar solicitudes de certificado en una variedad de formas y generar CRL. También mantiene una base de datos de texto de certificados emitidos y su estado.
Con openssl ca firmamos la solicitud anterior solicitud_sub.pem, especificamos el archivo de configuración mediante -config, se guardará con el nombre cacertsub.pem. Además añadimos el argumento -extensions v3_ca,extensiones que se agregarán cuando se emita el certificado.

```
patri@patri:~/Escritorio/CA$ openssl ca -in solicitud_sub.pem -config openssl.cn
f -extensions v3_ca -out cacertsub.pem
Using configuration from openssl.cnf
Enter pass phrase for /home/patri/Escritorio/CA/private/cakey.pem:
Check that the request matches the signature
Signature ok
Certificate Details:
        Serial Number: 4097 (0x1001)
        Validity
            Not Before: Nov 23 09:20:10 2018 GMT
            Not After : Nov 23 09:20:10 2019 GMT
        Subject:
            countryName               = ES
            stateOrProvinceName       = Granada
            organizationName          = SPSI
            organizationalUnitName    = SPSI-CAs
            commonName                = SPSI-CAs
        X509v3 extensions:
            X509v3 Subject Key Identifier:
                DA:C1:ED:2D:17:67:2C:87:CE:50:42:5A:26:9C:B8:6E:28:C2:E3:E8
            X509v3 Authority Key Identifier:
                keyid:A9:B5:10:CC:D5:55:17:2F:BA:5E:D5:9E:F0:57:C5:D7:8F:88:F4:F
8

            X509v3 Basic Constraints:
                CA:TRUE
Certificate is to be certified until Nov 23 09:20:10 2019 GMT (365 days)
Sign the certificate? [y/n]:y


1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
```

Figura 2.11: Certificado CA subordinada

El certificado cacertsub.pem lo renombraremos a cacert.pem para que sea correcto en cuanto al archivo de configuración de la CA subordinada, además de ubicarlo en la carpeta certs de la subordinada.
Visualizamos el contenido del certificado para la CA subordinada mediante el siguiente comando:

```
patri@patri:~/Escritorio/SUBCA$ openssl x509 -in certs/cacert.pem -noout -text
```

Figura 2.12: Certificado CA subordinada

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 4097 (0x1001)
    Signature Algorithm: sha256WithRSAEncryption
        Issuer: C=ES, ST=Granada, L=Granada, O=SPSI, OU=SPSI-CAr, CN=SPSI-CAr
        Validity
            Not Before: Nov 23 09:20:10 2018 GMT
            Not After : Nov 23 09:20:10 2019 GMT
        Subject: C=ES, ST=Granada, O=SPSI, OU=SPSI-CAs, CN=SPSI-CAs
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                Modulus:
                    00:ba:4c:16:5b:c2:2d:aa:d9:12:bd:9c:28:0f:9e:
                    9b:0c:35:72:0a:ab:dc:e4:85:ae:03:6b:26:a3:ce:
                    82:a0:45:87:ce:0d:25:ba:9d:98:ee:76:07:17:57:
                    6c:80:1f:1c:12:50:af:3a:b9:cc:f1:07:6f:68:17:
                    e9:4e:6d:d0:66:a9:12:83:c9:1b:a3:d7:5b:9c:5e:
                    87:99:94:b7:60:a6:85:7d:9a:f9:95:cf:3e:32:a9:
                    4b:71:68:40:2c:5e:ab:6c:02:19:4f:04:c4:62:48:
                    fa:85:1b:f8:4a:c8:3a:56:fe:ab:53:6f:17:fc:03:
                    92:37:43:71:ae:ab:70:59:19:96:a8:e4:a0:a7:45:
                    05:e4:b6:92:ad:5d:c7:bf:80:da:81:9b:58:fe:8f:
                    ab:88:26:5f:42:18:bd:cc:63:0e:b1:e3:af:3a:1f:
                    e4:01:ed:b4:fd:c0:cc:59:1e:43:45:c5:89:15:a1:
                    11:90:e1:dd:d9:28:59:e7:98:15:33:da:05:df:13:
                    ce:0d:b3:01:ae:8d:fa:31:82:31:38:29:3f:30:f7:
                    06:59:5b:5b:69:0c:25:13:68:33:bc:b0:b1:b6:81:
                    8d:20:18:6f:5c:e8:55:a8:f5:d9:00:b6:28:48:90:
                    d1:18:81:5f:3d:99:39:5d:b5:14:23:b3:8d:c1:cc:
                    7e:bd
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Subject Key Identifier:
                DA:C1:ED:2D:17:67:2C:87:CE:50:42:5A:26:9C:B8:6E:28:C2:E3:E8
            X509v3 Authority Key Identifier:
                keyid:A9:B5:10:CC:D5:55:17:2F:BA:5E:D5:9E:F0:57:C5:D7:8F:88:F4:F8
```

Figura 2.13: Certificado CA subordinada

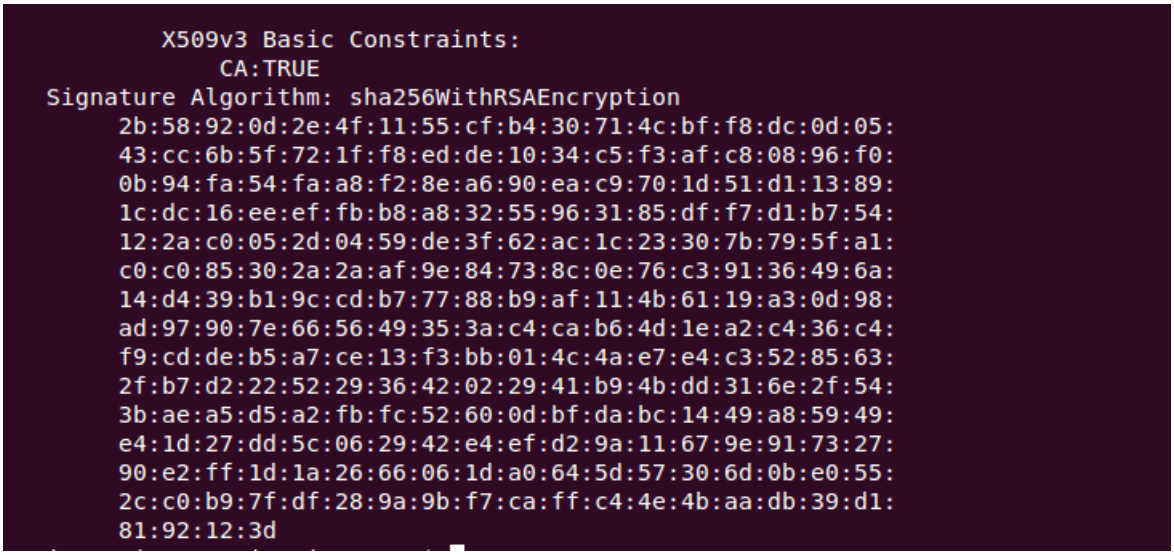

Figura 2.14: Certificado CA subordinada

3. Cread una solicitud de certificado que incluya la generación de claves en la misma.Mostrad los valores junto con el archivo

En los ejercicios anteriores hemos creado la clave y certificado en comandos separados, pero también existe la posibilidad de crear certificados incluyendo la generación de claves en el mismo comando. Para ello usaremos el comando **openssl req**. En vez de pasarle una clave ya creada, la creamos mediante -newkey rsa:2048 para que genere las claves RSA de 2048 bit guardandolas mediante el argumento -keyout en el directorio private. Además guardaremos la solicitud llamada solicitud_ejercicio3.pem en la raíz de la carpeta SUBCA, y le especificaremos el archivo de configuración mediante -config.

```
patri@patri:~/Escritorio/SUBCA$ openssl req -newkey rsa:2048 -keyout private/key
y.pem -out solicitud_ejercicio3.pem -config openssl.cnf
```

Figura 3.1: Solicitud certificado con generación de claves en la misma.

```
Generating a 2048 bit RSA private key
...........+++
...................+++
writing new private key to 'private/keyy.pem'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:ES
State or Province Name (full name) [Some-State]:Granada
Locality Name (eg, city) []:Granada
Organization Name (eg, company) [Internet Widgits Pty Ltd]:SPSI
Organizational Unit Name (eg, section) []:SPSI-CAs
Common Name (e.g. server FQDN or YOUR name) []:SPSI-CAs
Email Address []:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
```

Figura 3.2: Solicitud certificado con generación de claves en la misma.

Visualizamos el contenido de la solicitud.

```
patri@patri:~/Escritorio/SUBCA$ openssl  req -verify -in solicitud_ejercicio3.pe
m -text -noout
verify OK
Certificate Request:
    Data:
        Version: 0 (0x0)
        Subject: C=ES, ST=Granada, L=Granada, O=SPSI, OU=SPSI-CAs, CN=SPSI-CAs
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                Modulus:
                    00:c4:21:ba:d2:0f:82:cd:67:3a:55:d6:1d:83:90:
                    3f:5e:16:e1:ed:90:e6:8e:bd:18:0a:67:c2:0f:8d:
                    0f:ab:7b:b5:cc:0c:38:8d:87:e3:05:00:50:c7:40:
                    8a:b5:2b:58:c9:29:1a:6f:e5:d9:64:ca:5d:f8:b9:
                    38:f4:9b:e9:c6:fc:5c:7c:04:2a:2e:34:c9:37:9e:
                    c5:7f:2f:df:c4:8e:59:23:ab:6f:21:91:8d:2b:de:
                    00:e6:cc:3a:10:f7:b3:d2:e3:d3:09:39:14:fe:8a:
                    49:52:77:08:52:fe:04:75:4f:e4:79:4e:48:75:0a:
                    15:85:4d:1c:4f:cd:38:42:6d:8d:3b:c7:13:8b:e9:
                    36:6a:96:ff:23:0b:6d:89:10:31:76:37:01:32:34:
                    2b:54:53:00:03:a0:7d:4c:f2:d6:18:27:76:5f:fd:
                    8d:59:bb:b7:20:f1:2d:c7:d0:91:dd:5e:67:5c:0a:
                    6d:cc:32:e1:48:c4:0e:a8:33:5e:16:1f:41:b0:ad:
                    fd:87:a5:db:c3:3f:6a:f0:4f:cf:fc:eb:b9:e3:da:
                    ea:cb:d2:8c:1d:e8:50:53:55:fd:f6:5d:c0:e0:63:
                    d6:78:47:dc:b2:e1:69:d1:c4:16:07:8b:af:34:25:
                    e6:00:d3:20:89:e6:e8:8a:44:b1:18:a5:f0:02:16:
                    94:cf
                Exponent: 65537 (0x10001)
        Attributes:
            a0:00
    Signature Algorithm: sha256WithRSAEncryption
         06:9f:9e:bc:98:c0:55:01:a4:53:fc:aa:2f:39:30:56:a0:a4:
         0d:f1:2d:de:d7:d3:fc:39:fb:08:67:0c:44:5e:ac:4a:67:d8:
         b6:70:3b:93:bc:fc:18:be:37:bc:65:3d:ce:1d:89:2d:40:10:
         8c:13:c9:88:17:49:3b:10:43:8f:4f:1f:ba:a8:4c:21:cc:99:
         10:e8:4f:41:77:f0:f8:30:0c:4e:66:64:2c:04:2c:05:f0:bc:
         8e:f4:a6:a0:bb:8d:4a:9e:0a:f9:c2:1b:40:07:91:e9:cd:09:
         00:b2:80:7d:16:ac:84:b0:c8:62:3e:2c:6f:4b:59:3a:e4:19:
         f0:df:fd:63:2c:98:16:fc:1e:99:78:9d:9c:70:8a:70:16:d6:
         31:e1:2f:9b:0e:c9:96:37:6c:38:1e:55:e9:69:63:31:04:9c:
         9c:64:15:9f:8f:a8:49:1c:55:45:57:14:a1:91:16:b9:55:ce:
         0a:62:7b:87:48:d7:49:2a:5e:3a:17:ad:ee:64:e5:d6:18:c3:
         6d:af:e4:de:1f:77:39:c3:4a:6d:5e:96:40:ea:81:df:82:cf:
         0f:e1:ea:51:50:7c:ef:f6:40:04:7c:b6:59:d8:fa:95:98:27:
         fa:43:50:da:3c:00:15:f6:df:28:bb:68:8f:9f:8b:88:51:49:
         41:58:fa:0b
```

Figura 3.3: Contenido solicitud certificado ejercicio3

4. Cread un certificado para la solicitud anterior empleando la CA subordinada. Mostrad el archivo y sus valores

Utilizamos el comando openssl ca para crear el certficado de la solicitud del ejercicio anterior. Le indicamos el archivo de solicitud mediante el argumento -in, el archivo de configuración, y mediante -out guardaremos el certificado, al que llamaremos certificado_ejercicio3.pem.

```
patri@patri:~/Escritorio/SUBCA$ openssl ca -in solicitud_ejercicio3.pem -out cer
tificado_ejercicio3.pem -config openssl.cnf
```

Figura 4.1: Certificado para la solicitud del ejercicio 3

```
Enter pass phrase for /home/patri/Escritorio/SUBCA/private/cakey.pem:
Check that the request matches the signature
Signature ok
Certificate Details:
        Serial Number: 4097 (0x1001)
        Validity
            Not Before: Nov 24 03:23:05 2018 GMT
            Not After : Nov 24 03:23:05 2019 GMT
        Subject:
            countryName               = ES
            stateOrProvinceName       = Granada
            organizationName          = SPSI
            organizationalUnitName    = SPSI-CAs
            commonName                = SPSI-CAs
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            Netscape Comment:
                OpenSSL Generated Certificate
            X509v3 Subject Key Identifier:
                2D:42:A8:CC:A1:7C:94:75:0D:EC:2A:52:E7:EA:58:E4:4C:CC:1C:74
            X509v3 Authority Key Identifier:
                keyid:DA:C1:ED:2D:17:67:2C:87:CE:50:42:5A:26:9C:B8:6E:28:C2:E3:E
8

Certificate is to be certified until Nov 24 03:23:05 2019 GMT (365 days)
Sign the certificate? [y/n]:y


1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
```

Figura 4.2: Certificado para la solicitud del ejercicio 3

Contenido del certificado.

```
patri@patri:~/Escritorio/SUBCA$ openssl x509 -in certificado_ejercicio3.pem -noo
ut -text
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 4097 (0x1001)
    Signature Algorithm: sha256WithRSAEncryption
        Issuer: C=ES, ST=Granada, O=SPSI, OU=SPSI-CAs, CN=SPSI-CAs
        Validity
            Not Before: Nov 24 03:23:05 2018 GMT
            Not After : Nov 24 03:23:05 2019 GMT
        Subject: C=ES, ST=Granada, O=SPSI, OU=SPSI-CAs, CN=SPSI-CAs
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                Modulus:
                    00:c4:21:ba:d2:0f:82:cd:67:3a:55:d6:1d:83:90:
                    3f:5e:16:e1:ed:90:e6:8e:bd:18:0a:67:c2:0f:8d:
                    0f:ab:7b:b5:cc:0c:38:8d:87:e3:05:00:50:c7:40:
                    8a:b5:2b:58:c9:29:1a:6f:e5:d9:64:ca:5d:f8:b9:
                    38:f4:9b:e9:c6:fc:5c:7c:04:2a:2e:34:c9:37:9e:
                    c5:7f:2f:df:c4:8e:59:23:ab:6f:21:91:8d:2b:de:
                    00:e6:cc:3a:10:f7:b3:d2:e3:d3:09:39:14:fe:8a:
                    49:52:77:08:52:fe:04:75:4f:e4:79:4e:48:75:0a:
                    15:85:4d:1c:4f:cd:38:42:6d:8d:3b:c7:13:8b:e9:
                    36:6a:96:ff:23:0b:6d:89:10:31:76:37:01:32:34:
                    2b:54:53:00:03:a0:7d:4c:f2:d6:18:27:76:5f:fd:
                    8d:59:bb:b7:20:f1:2d:c7:d0:91:dd:5e:67:5c:0a:
                    6d:cc:32:e1:48:c4:0e:a8:33:5e:16:1f:41:b0:ad:
                    fd:87:a5:db:c3:3f:6a:f0:4f:cf:fc:eb:b9:e3:da:
                    ea:cb:d2:8c:1d:e8:50:53:55:fd:f6:5d:c0:e0:63:
                    d6:78:47:dc:b2:e1:69:d1:c4:16:07:8b:af:34:25:
                    e6:00:d3:20:89:e6:e8:8a:44:b1:18:a5:f0:02:16:
                    94:cf
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            Netscape Comment:
                OpenSSL Generated Certificate
    Signature Algorithm: sha256WithRSAEncryption
         69:b0:7b:ab:37:cc:33:d7:88:a7:4e:87:83:ff:b7:16:29:33:
         8e:fe:fe:ae:94:56:2e:a4:41:e9:22:d3:ba:95:23:11:94:e5:
         53:80:42:72:7f:64:24:2c:31:20:25:50:bb:b2:21:93:9e:b3:
         24:9f:ee:2d:71:47:64:c3:e7:46:89:b7:28:96:be:9b:a5:0d:
         a5:4f:30:8b:ef:64:de:ea:ba:a1:ee:15:c0:08:9d:f4:d4:a4:
         b8:9a:ea:36:fb:d7:ca:62:d5:b3:21:fc:37:ca:8d:42:02:7e:
         e8:32:22:d4:41:d6:77:09:2e:ee:f1:96:ae:57:eb:f8:71:3a:
         2a:67:85:38:87:63:3c:d6:a8:3e:07:fb:65:33:4d:26:1c:41:
         27:34:01:f4:10:ec:9e:18:ef:00:0a:73:03:72:47:84:62:0f:
         fa:7c:01:52:43:67:fa:ea:f7:5a:64:49:b3:bf:29:a6:bb:7b:
         21:2a:27:19:37:35:33:3d:e6:fc:77:d7:26:74:b4:e6:4a:31:
         1a:f8:d0:1b:74:f0:12:dd:61:ea:aa:79:fc:10:c1:1f:37:9d:
         dd:e3:98:62:b2:a9:94:08:9a:42:10:45:86:35:2d:58:af:bd:
         56:4b:04:ce:87:3d:fa:7b:69:f2:1b:30:49:3a:e6:36:75:b8:
         96:1c:06:71
```

Figura 4.3: Certificado

5. Cread una solicitud de certificado para cualquiera de las claves que habéis generado en las prácticas anteriores, excepto las RSA. Mostrad el archivo y el valor de la solicitud.

Para este ejercicio he usado el archivo patriciaECkey.pem, generado en las prácticas anteriores. Éste contiene la pareja de claves EC.
Creamos la solicitud mediante el comando openssl req, al igual que en el ejercicio3, con la diferencia que en este caso le pasamos el archivo que he mencionado anteriormente patriciaECkey.pem. La solicitud se guardará con el nombre solicitudEj5.pem en la carpeta SUBCA.

```
patri@patri:~/Escritorio/SUBCA$ openssl req -config openssl.cnf -key patriciaECk
ey.pem -new -out solicitudEj5.pem
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:ES
State or Province Name (full name) [Some-State]:Granada
Locality Name (eg, city) []:Granada
Organization Name (eg, company) [Internet Widgits Pty Ltd]:SPSI
Organizational Unit Name (eg, section) []:SPSI-CAs
Common Name (e.g. server FQDN or YOUR name) []:SPSI-CAs
Email Address []:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
```

Figura 5.1: solicitud

Visualizamos la solicitud:

```
patri@patri:~/Escritorio/SUBCA$ openssl  req -verify -in solicitudEj5.pem -text
-noout
verify OK
Certificate Request:
    Data:
        Version: 0 (0x0)
        Subject: C=ES, ST=Granada, L=Granada, O=SPSI, OU=SPSI-CAs, CN=SPSI-CAs
        Subject Public Key Info:
            Public Key Algorithm: id-ecPublicKey
                Public-Key: (192 bit)
                pub:
                    04:40:88:72:cb:3d:98:44:5c:15:a1:94:b3:19:3c:
                    21:cd:7d:df:15:72:a5:c3:58:13:70:db:99:4e:69:
                    97:ea:b7:8a:75:31:2f:60:fc:ec:90:08:53:96:9f:
                    b0:1e:f7:34
                ASN1 OID: prime192v1
        Attributes:
            a0:00
    Signature Algorithm: ecdsa-with-SHA256
         30:35:02:19:00:c2:66:ee:e2:be:00:f8:81:90:6d:fb:9f:d4:
         e3:03:2c:bd:dc:55:f4:5b:9e:5d:83:02:18:17:f0:33:f4:2e:
         b5:e8:e1:00:c2:a2:d2:df:6b:74:64:5e:f1:75:cd:ee:88:bc:
         6e
```

Figura 5.2: solicitud ejercicio 5

6. Cread un certificado para la solicitud anterior utilizando la CA subordinada. Mostrad el archivo y los valores del certificado

Para crear el certificado, usamos el comando openssl ca, al igual que en el ejercicio 4. Que guardaremos como certificadoEj6.pem

```
patri@patri:~/Escritorio/SUBCA$ openssl ca -in solicitudEj5.pem -out certificado
Ej6.pem -config openssl.cnf
Using configuration from openssl.cnf
Enter pass phrase for /home/patri/Escritorio/SUBCA/private/cakey.pem:
Check that the request matches the signature
Signature ok
Certificate Details:
        Serial Number: 4097 (0x1001)
        Validity
            Not Before: Nov 23 10:18:35 2018 GMT
            Not After : Nov 23 10:18:35 2019 GMT
        Subject:
            countryName               = ES
            stateOrProvinceName       = Granada
            organizationName          = SPSI
            organizationalUnitName    = SPSI-CAs
            commonName                = SPSI-CAs
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            Netscape Comment:
                OpenSSL Generated Certificate
            X509v3 Subject Key Identifier:
                0C:DC:E8:6B:0F:0C:14:4B:53:B8:68:AC:40:62:74:A9:8B:5A:3C:43
            X509v3 Authority Key Identifier:
                keyid:DA:C1:ED:2D:17:67:2C:87:CE:50:42:5A:26:9C:B8:6E:28:C2:E3:E
8

Certificate is to be certified until Nov 23 10:18:35 2019 GMT (365 days)
Sign the certificate? [y/n]:y


1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
```

Figura 6.1: Comando para crear certificado ejercicio 6

Visualizamos el contenido del certificado:

```
patri@patri:~/Escritorio/SUBCA$ openssl x509 -in certificadoEj6.pem -noout -text

Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 4097 (0x1001)
    Signature Algorithm: sha256WithRSAEncryption
        Issuer: C=ES, ST=Granada, O=SPSI, OU=SPSI-CAs, CN=SPSI-CAs
        Validity
            Not Before: Nov 23 10:18:35 2018 GMT
            Not After : Nov 23 10:18:35 2019 GMT
        Subject: C=ES, ST=Granada, O=SPSI, OU=SPSI-CAs, CN=SPSI-CAs
        Subject Public Key Info:
            Public Key Algorithm: id-ecPublicKey
                Public-Key: (192 bit)
                pub:
                    04:40:88:72:cb:3d:98:44:5c:15:a1:94:b3:19:3c:
                    21:cd:7d:df:15:72:a5:c3:58:13:70:db:99:4e:69:
                    97:ea:b7:8a:75:31:2f:60:fc:ec:90:08:53:96:9f:
                    b0:1e:f7:34
                ASN1 OID: prime192v1
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            Netscape Comment:
                OpenSSL Generated Certificate
            X509v3 Subject Key Identifier:
                0C:DC:E8:6B:0F:0C:14:4B:53:B8:68:AC:40:62:74:A9:8B:5A:3C:43
            X509v3 Authority Key Identifier:
                keyid:DA:C1:ED:2D:17:67:2C:87:CE:50:42:5A:26:9C:B8:6E:28:C2:E3:E
8

    Signature Algorithm: sha256WithRSAEncryption
         3b:9f:27:9b:31:f8:5e:ef:f9:95:2e:90:7c:fc:c2:93:ed:96:
         67:2c:1a:2a:2f:6a:a1:83:18:3e:13:b8:f6:a0:03:73:6d:44:
         46:30:d9:e2:42:a6:b5:d3:f2:59:59:02:84:61:f5:ee:16:23:
         9c:ed:a1:7b:e6:48:14:e8:af:48:3e:4e:4d:cc:68:a0:9c:d2:
         8c:13:30:66:d5:26:a6:04:c7:ad:6f:b9:63:38:28:11:af:93:
         54:9e:9a:8d:85:34:41:da:67:fe:e9:37:d6:a3:f8:19:b8:72:
         27:63:0c:b0:2c:1f:bf:36:92:81:65:40:16:d4:12:f8:1e:68:
         f1:1e:24:b2:06:c0:98:ce:50:92:2a:9b:40:02:b7:4c:dc:fc:
         b4:1e:00:92:ba:b0:d2:d2:94:23:8f:e0:fe:95:78:19:20:3b:
         97:8c:6c:67:cc:59:12:36:a6:6e:46:e1:ed:13:55:98:52:f1:
         da:e0:71:4b:33:c5:29:3a:98:8b:a7:ca:10:f5:c1:eb:32:58:
         3b:ba:ca:2b:d6:d7:14:36:cb:43:6a:fc:bf:66:49:af:ac:76:
         72:1b:b5:fa:b2:25:f8:ad:1f:77:61:14:e4:3c:e4:d8:f8:57:
         f9:be:b4:ee:da:a0:80:4d:4e:db:33:de:04:57:0d:19:17:f8:
         8d:70:da:86
```

Figura 6.2: Contenido del certificado ejercicio 6

7. Referencias

Creación de CA y subordinada
https://itsecworks.com/2010/11/22/create-your-own-ca-or-root-ca-subordinate-ca/
Solicitud y generación de certificados
https://www.openssl.org/docs/manmaster/man1/req.html
https://www.openssl.org/docs/man1.0.2/apps/ca.html