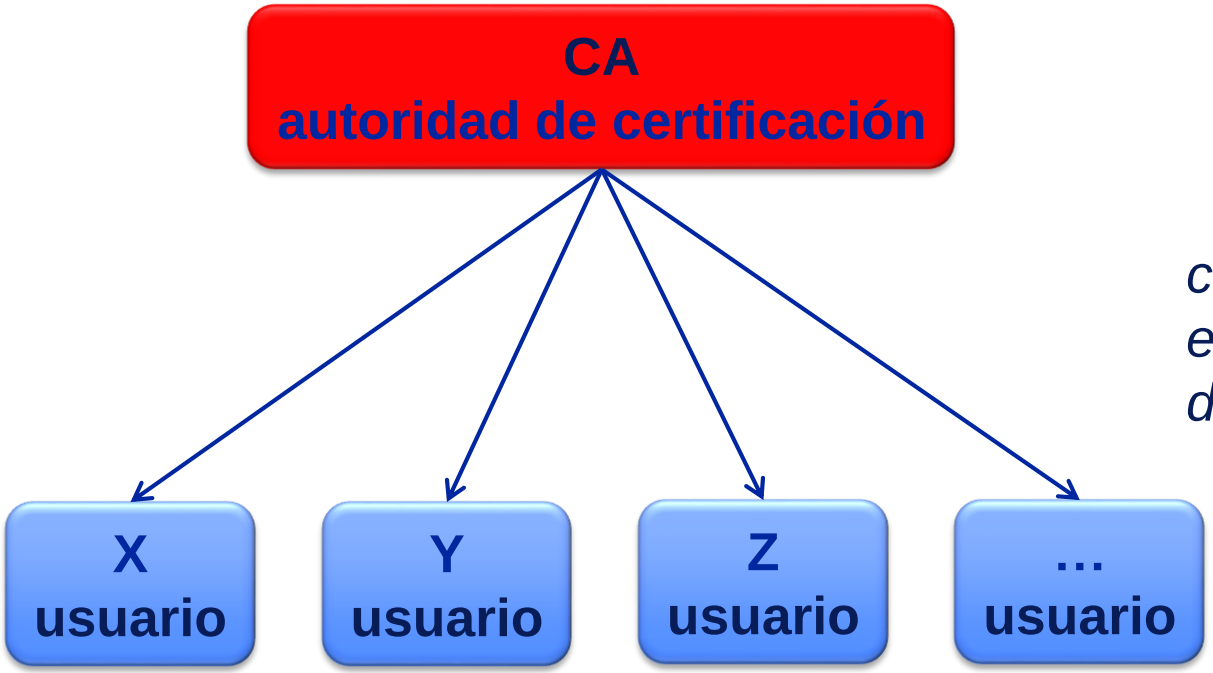
Autoridad de Certificación usando OpenSSL

José A. Mañas < http://www.dit.upm.es/~pepe/>
Dep. de Ingeniería de Sistemas Telemáticos
E.T.S. Ingenieros de Telecomunicación
Universidad Politécnica de Madrid

3 de octubre de 2016

arquitectura



cada cajita es un ordenador diferente

OpenSSL

1. descargar e instalar openssl
   - https://www.openssl.org/
2. prepare un directorio de trabajo
   - debe estar en las carpetas del alumno, con todos los derechos de lectura y escritura
3. preparar el fichero de configuración (o descargarlo)
   - http://www.dit.upm.es/~pepe/doc/seg4/ca

directorios
- CA

ficheros
- CA / openssl.cnf
- CA / serial
- CA / index.txt

ficheros

- CA / openssl.cnf
  - se descarga de la web y se personaliza
- CA / serial
  - se crea con un número que será el ID del primer certificado generado
  - por ejemplo: 1000
- CA /index.txt
  - se crea vacío (0 bytes)

claves de la CA

- creamos una clave RSA de 4096 bits

```
openssl genpkey -out ca_sec.pem -aes256 -algorithm rsa -pkeyopt
rsa_keygen_bits:4096
```

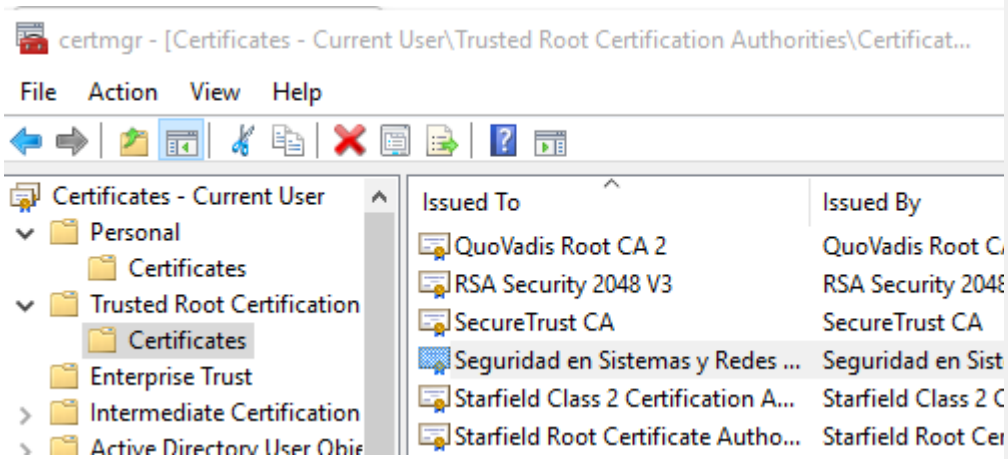
- extraemos la parte pública

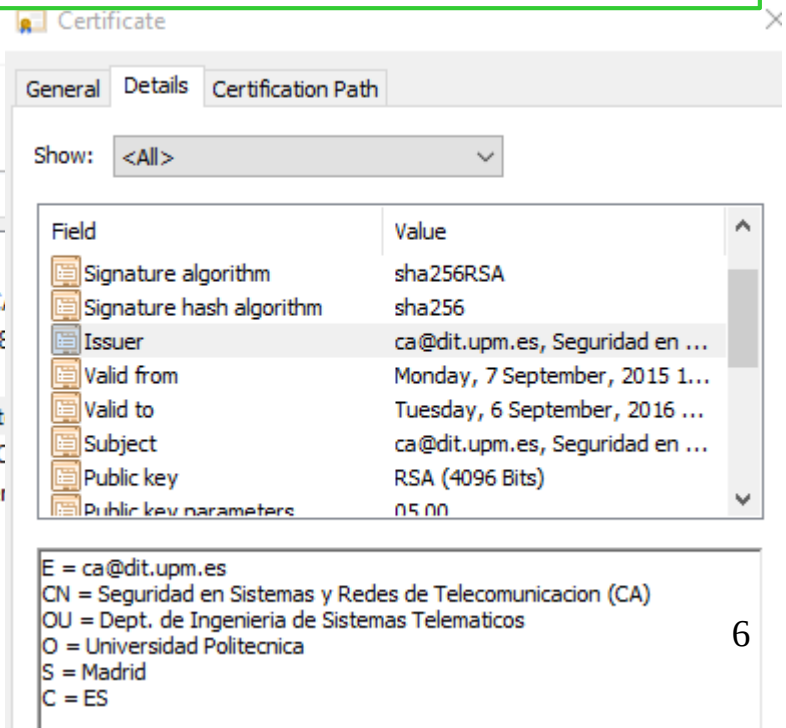
```
openssl rsa -in ca_sec.pem -pubout -out ca_pub.pem
```

- ficheros
  - ca_sec.pem parte secreta
  - ca_pub.pem parte pública

certificado de la CA

- genera un [auto] certificado
  - openssl req -new -config openssl.cnf -days 365 -key ca_sec.pem -x509 -out ca_cert.pem
- cambio de formato
  - openssl x509 -outform der -in ca_cert.pem -out ca.crt





usuario X

- generación de claves

```
openssl genpkey -out X_sec.pem -aes256 -algorithm rsa -pkeyopt
rsa_keygen_bits:2048
```

- extraemos la parte pública

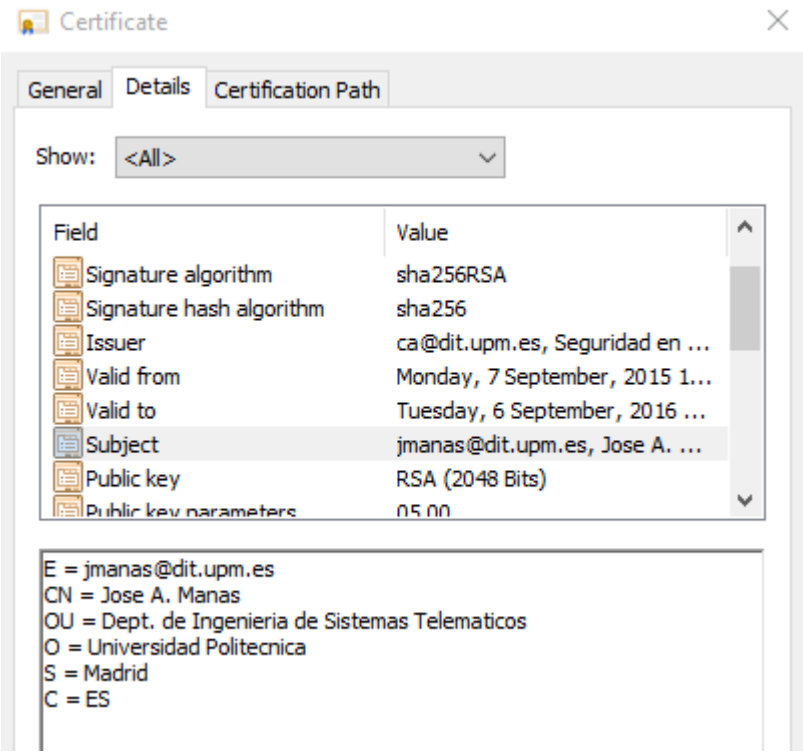
```
openssl rsa -in X_sec.pem -pubout -out X_pub.pem
```

- solicitud (sec → csr)

  - openssl req -new -config openssl.cnf -days 365 -key X_sec.pem -out X.csr

la CA certifica

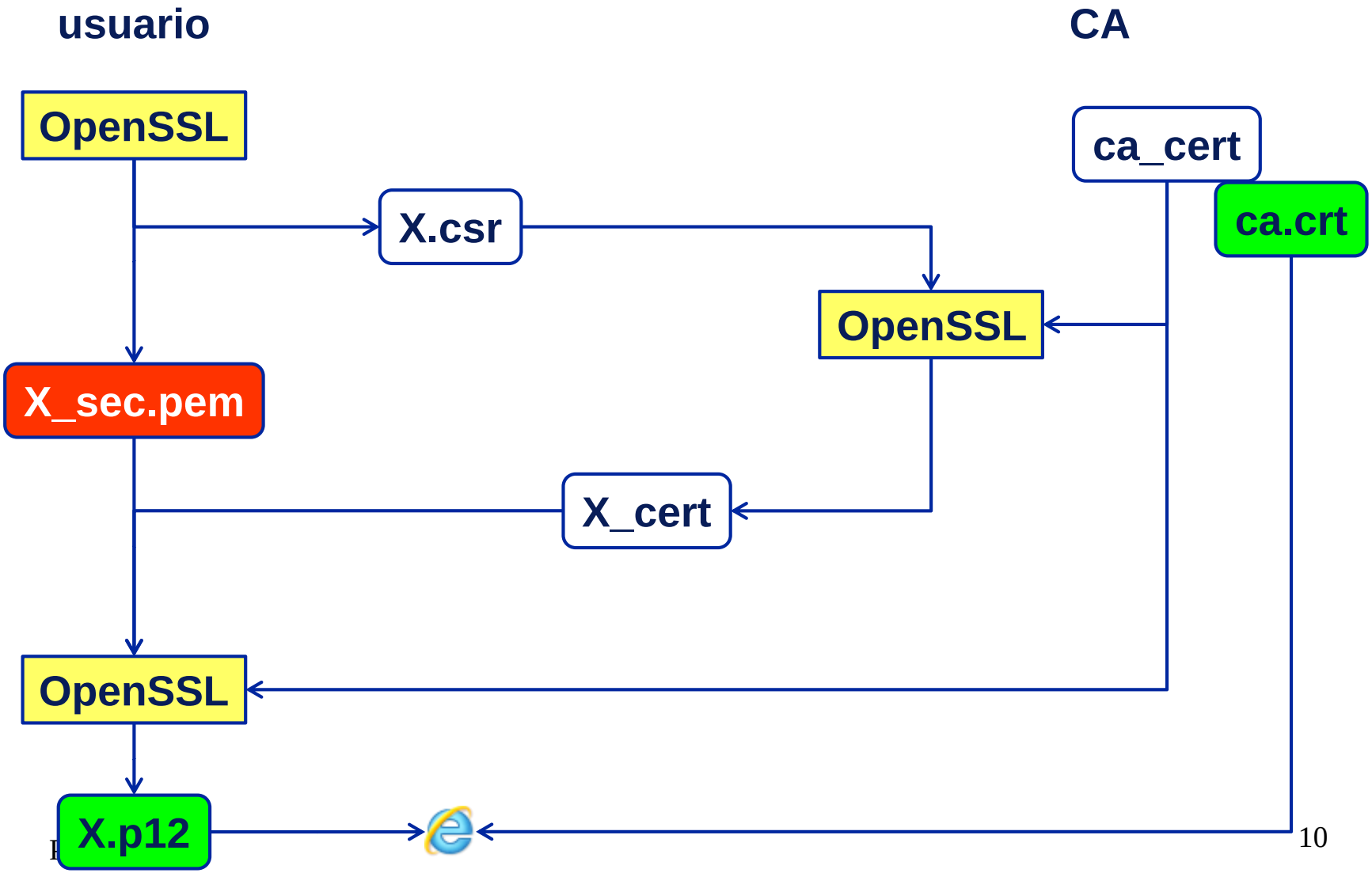
- CA: genera el certificado (csr → cert)
  - openssl ca -config openssl.cnf -in X.csr -out X_cert.pem



altogether: p12

- claves
  - openssl pkcs12 -export -out X.p12 -inkey X_sec.pem -in X_cert.pem -certfile ca_cert.pem
  - exportar a p12 significa crear un paquete que agrupa la clave privada y los certificados del usuario (el propio y los que le avalan)
  - por contener una clave privada, se protege con una ontraseña que se pedirá cada vez que se quiera usar el p12
    (por ejemplo para cargarlo en el registro de certificados)
- cambio de formato
  - openssl x509 -outform der -in X_cert.pem -out X.crt

pasos y ficheros



dit

ordenadores y ficheros

CA

- openssl.cnf
- serial
- index.txt
- ca_cert.pem = ca.crt
- ca_sec.pem

Z

Y

X

- openssl.cnf
- ca_cert.pem = ca.crt
- X_sec.pem
- X.csr
- X_cert.pem = X.crt
- X.p12

dit para ver …

- una clave privada
  - openssl rsa -in X_sec.pem -noout –text
- una clave pública
  - openssl rsa -pubin -in X_pub.pem -noout -text
- una solicitud de certificación
  - openssl req -in X.csr -noout -text
- un certificado
  - openssl x509 -in X_cert.pem -text -noout
  - openssl x509 –inform der -in X.crt -text -noout

otras funciones

- revocaciones de certificados
- certificado de servidor SSL
- certificado de cliente SSL
- cifra de ficheros de texto (usando certificado)
- descifrar (usando la clave secreta)

revocaciones

- para revocar un certificado
  - openssl ca -config openssl.cnf -revoke X_cert.pem
- para generar la lista de revocados
  - openssl ca -config openssl.cnf -gencrl -out crl.pem
- cambio de formato
  - openssl crl -in crl.pem -outform der -out ca.crl
- para verlo
  - openssl crl -in crl.pem -noout -text
  - openssl crl -inform der -in ca.crl -noout -text
- … y hay que publicarla en un lugar accesible
  - http://www.dit.upm.es/~pepe/doc/seg4/ca.crl

servidor https (ssl)

- openssl genpkey -out server_sec.pem -algorithm rsa -pkeyopt rsa_keygen_bits:2048
- openssl rsa -in server_sec.pem -pubout -out server_pub.pem
- openssl req -nodes -new -config openssl.cnf -extensions ssl_server -days 365 -key server_sec.pem -out server_csr.pem
- openssl ca -config openssl.cnf -extensions ssl_server -in server_csr.pem -out server_cert.pem

```
[ ssl_server ]

basicConstraints = CA:FALSE
subjectKeyIdentifier = hash
authorityKeyIdentifier = keyid,issuer:always
keyUsage = digitalSignature, keyEncipherment
extendedKeyUsage = serverAuth
```

cliente https (ssl)

- openssl genpkey -out client_sec.pem -algorithm rsa -pkeyopt rsa_keygen_bits:2048
- openssl rsa -in client_sec.pem -pubout -out client_pub.pem
- openssl req -nodes -new -config openssl.cnf -extensions ssl_client -days 365 -key client_sec.pem -out client_csr.pem
- openssl ca -config openssl.cnf -extensions ssl_client -in client_csr.pem -out client_cert.pem

```
[ ssl_client ]

basicConstraints = CA:FALSE
subjectKeyIdentifier = hash
authorityKeyIdentifier = keyid,issuer:always
keyUsage = digitalSignature, keyEncipherment
extendedKeyUsage = clientAuth
```

cifra de ficheros

- se genera una clave simétrica K para cifrar el mensaje
- se cifra la clave K con la parte pública del certificado
- se recupera K con la parte secreta de la clave

1. openssl smime -encrypt -aes256
   -in file.txt -out file.txt.smime lola_cert.pem
2. openssl smime -decrypt
   -in file.txt.smime -inkey lola_sec.pem -out decrypted

Anexos

- formatos de ficheros
- fichero de configuración

sobre formatos

- formatos más habituales: PEM y DER
- PEM – privacy enhanced mail
  - codificado ASCII (base-64)
  - RFC 1421
- DER – Distinguished Encoding Rules
  - binario (ASN.1)
  - ITU-T X.690

ejemplo formato PEM

```
$ head ca_cert.pem
-----BEGIN CERTIFICATE-----
MIIEZDCCA0ygAwIBAgIJAMHLR+JWHl2iMA0GCSqGSIb3DQEBCwUAMIGhMQswCQYD
VQQGEwJFUzEPMA0GA1UECAwGTWFkcmlkMSAwHgYDVQQKDBdVbml2ZXJzaWRhZCBQ
b2xpdGVjbmljYTE0MDIGA1UECwwrRGVwdC4gZGUgSW5nZW5pZXJpYSBkZSBTaXN0
ZW1hcyBUZWxlbWF0aWNvczELMAkGA1UEAwwCQ0ExHDAaBgkqhkiG9w0BCQEWDWNh
QGRpdC51cG0uZXMwHhcNMTYwOTA3MTc1MjM3WhcNMjYwOTA1MTc1MjM4WjCBoTEL
MAkGA1UEBhMCRVMxDzANBgNVBAgMBk1hZHJpZDEgMB4GA1UECgwXVW5pdmVyc2lk
```

extensiones típicas

dit

- certificados
  - .crt .cer
- claves privadas
  - .p12 .pfx
- solicitudes de certificados
  - .csr .p10
- revocaciones
  - .crl

todos estos ficheros
suele estar codificados con DER
excepto CSR que suele ser PEM

dit openssl.cnf

- configura la operación de openssl
- consta de parejas “clave=valor” agrupadas en [secciones]

[ca]
parámetros para “openssl ca”
[req]
parámetros para “openssl req”

atributos varios
[CA_default]
[policy_match]
[req_distinguished_name]
[req_attributes]
[usr_cert]
[self_cert_extensions]

dit

openssl.cnf

```
dir = .

[ ca ]

default_ca = CA_default
```

```
[ CA_default ]

serial = $dir/serial
database= $dir/index.txt
new_certs_dir = $dir

certificate = $dir/ca_cert.pem
private_key = $dir/ca_sec.pem

certs= $dir
unique_subject = no
x509_extensions = usr_cert
name_opt = ca_default
cert_opt = ca_default
copy_extensions = copy

default_days = 365
default_md = sha256
preserve = no
policy = policy_match
```

```
[ policy_match ]

countryName = match
stateOrProvinceName = match
localityName = optional
organizationName = match
organizationalUnitName = optional
commonName = supplied
emailAddress = optional
```

openssl.cnf

```
[ usr_cert ]

basicConstraints = CA:FALSE
keyUsage = nonRepudiation, digitalSignature, keyEncipherment
extendedKeyUsage = emailProtection
nsComment = "OpenSSL Generated Certificate"
subjectKeyIdentifier = hash
authorityKeyIdentifier = keyid,issuer
```

dit

openssl.cnf

```
[ req ]

default_bits = 2048
default_keyfile = key_sec.pem
distinguished_name = req_distinguished_name
attributes = req_attributes
x509_extensions = self_cert_extensions
string_mask = utf8only
```

```
[ req_attributes ]

challengePassword = A challenge password
challengePassword_min = 4
challengePassword_max = 20
```

```
[ self_cert_extensions ]

basicConstraints = CA:true
keyUsage = cRLSign, keyCertSign
subjectKeyIdentifier = hash
authorityKeyIdentifier = keyid:always,issuer
```

dit

openssl.cnf

```
[ req_distinguished_name ]

countryName = Country Name (2 letter code)
countryName_min = 2
countryName_max = 2
stateOrProvinceName = State or Province Name (full name)
localityName = Locality Name (eg, city)
0.organizationName = Organization Name (eg, company)
organizationalUnitName = Organizational Unit Name (eg, section)
commonName = Common Name (eg, YOUR name)
commonName_max = 64
emailAddress = Email Address
emailAddress_max = 64

countryName_default = ES
stateOrProvinceName_default = Madrid
0.organizationName_default = Universidad Politecnica
organizationalUnitName_default = Dept. de Ingenieria de Sistemas Telematicos
```

revocaciones

- openssl.cnf

```
[ CA_default ]

crl_dir= $dir
crlnumber = $crl_dir/crlnumber
crl = $crl_dir/crl.pem
default_crl_days= 30
crl_extensions  = crl_ext
```

```
[ usr_cert ]

crlDistributionPoints = URI:http://www.dit.upm.es/~pepe/doc/seg4/ca.crl
```

```
[ v3_ca ]

crlDistributionPoints = URI:http://www.dit.upm.es/~pepe/doc/seg4/ca.crl
```