GUÍA 6: PKI Y CERTIFICADOS DIGITALES

Ejercicio 1: CERTIFICADOS DIGITALES

Un certificado digital consiste en una clave pública y un identificador o nombre de usuario del dueño de la clave, firmado por una tercera parte confiable (autoridad certificante)

El primer paso para obtener un certificado es crear una solicitud de certificado. En dicha solicitud, habrá que incluir la clave privada y otros datos que identifiquen al usuario. Son campos de un nombre x500. Para ello, usar el comando **req:**

```
openssl req [-inform PEM|DER] [-outform PEM|DER] [-in filename] [-passin
arg] [-out filename] [-passout arg] [-text] [-pubkey] [-noout] [-verify] [-
modulus] [-new] [-rand file(s)] [-newkey rsa:bits] [-newkey dsa:file] [-
newkey alg:file] [-nodes] [-key filename] [-keyform PEM|DER] [-keyout
filename] [-[md5|sha1|md2|mdc2]] [-config filename] [-subj arg] [-
multivalue-rdn] [-x509] [-days n] [-set_serial n] [-asn1-kludge] [-newhdr]
[-extensions section] [-reqexts section] [-utf8] [-nameopt] [-batch] [-
verbose] [-engine id]
```

```
$ openssl req –new –key priv.pem –out solicitud.csr
```

Ejercicio 2:

Como aún no tenemos autoridad certificante, lo autocerficarás. Te certificarás a vos mismo haciendo:

```
$ openssl req –x509 –key priv.pem –in solicitud.csr –out autocertif.pem
```

Observa las diferencias entre el archivo solicitud.csr y autocertif.pem

Ejercicio 3:

1) Para crear una CA (autoridad certificante), será necesario en primer lugar que generes un par de claves privada y pública: **CApriv.key** y **CApub.key**

2) Luego, crea un archivo de texto llamado **CAconf1.cfg** con el siguiente contenido: (parámetros que se usarán para crear certificados digitales)

```
[ req ]
default_bits            = 1024
default_keyfile         = CApriv.key
distinguished_name      = req_distinguished_name
attributes              = req_attributes
x509_extensions         = v3_ca
dirstring_type          = nobmp

[ req_distinguished_name ]
countryName             = Identificador del Pais (2 letras)
countryName_default     = AR
countryName_min         = 2
countryName_max         = 2
localityName            = Localidad (ej., ciudad)
organizationalUnitName  = Nombre de unidad organizacional (ej., oficina)
commonName              = Nombre común (ej., TU nombre)
commonName_max          = 64
emailAddress            = direccion de correo electrónico
emailAddress_max        = 40

[ req_attributes ]
challengePassword       = Contraseña para "challenge"
challengePassword_min   = 4
challengePassword_max   = 20

[ v3_ca ]
subjectKeyIdentifier    = hash
authorityKeyIdentifier  = keyid:always, issuer:always
basicConstraints        = CA:true
```

y el archivo **CAconf2.cfg** (completa los campos de req_distinguished_name con los datos de quien será la autoridad certificante)

```
[ req ]
default_bits            = 1024
default_keyfile         = CApriv.key
distinguished_name      = req_distinguished_name
attributes              = req_attributes
prompt                  = no
output_password         = mipassword
x509_extensions         = v3_ca
dirstring_type          = nobmp

[ req_distinguished_name ]
C                       = AR
ST                      = Buenos Aires
L                       = Buenos Aires
O                       = Empresa Ficticia
OU                      = Oficina de SI
CN                      = Ana Arias
emailAddress            = ariasroigana@gmail.com

[ req_attributes ]
challengePassword       = Contraseña para "challenge"
challengePassword_min   = 4
challengePassword_max   = 20

[ v3_ca ]
subjectKeyIdentifier    = hash
authorityKeyIdentifier  = keyid:always, issuer:always
basicConstraints        = CA:true
```

3) Con estos archivos preparados, crear un **certificado de autoridad** con el siguiente comando:

```
openssl req –new –key CApriv.key –out ca.cer –config CAconf2.cfg –x509 –days 3650
```

Este certificado digital está autofirmado (por la misma CA). Tiene una duración de 10 años.

La autoridad certificante ya tiene clave privada (**CApriv.key**), clave pública (**CApub.key**) y certificado autofirmado (**ca.cer**). Ya está en condiciones de certificar otros certificados.

4) De manera análoga al ejercicio 6, se creará un requerimiento de certificado. Deberás tener las claves privada y pública del usuario (**USRpriv.key** y **USRpub.key**) Usa el archivo de configuración CAconf1.cfg y guarda el requerimiento como **req.pem**

5) Ahora procede a firmar el requerimiento y generar el certificado del usuario (**USRcert.cer**). Usar el comando x509:

```
openssl x509 [-inform DER|PEM|NET] [-outform DER|PEM|NET] [-keyform DER|PEM]
[-CAform DER|PEM] [-CAkeyform DER|PEM] [-in filename] [-out filename] [-
serial] [-hash] [-subject_hash] [-issuer_hash] [-subject] [-issuer] [-
nameopt option] [-email] [-ocsp_uri] [-startdate] [-enddate] [-purpose] [-
dates] [-modulus] [-fingerprint] [-alias] [-noout] [-trustout] [-clrtrust]
[-clrreject] [-addtrust arg] [-addreject arg] [-setalias arg] [-days arg] [-
set_serial n] [-signkey filename] [-x509toreq] [-req] [-CA filename] [-CAkey
filename] [-CAcreateserial] [-CAserial filename] [-text] [-C] [-md2|-md5|-
sha1|-mdc2] [-clrext] [-extfile filename] [-extensions section] [-engine id]
```

Colocar en todos los formatos la opción PEM, generarlo para una validez de 1 año, usando hash sha1, y la opcion –text para que lo cree en formato de texto. La opción –CA debe tener como argumento el certificado de la CA.

6) Observa el certificado obtenido (USRcert.cer). Toma nota del contenido del certificado. Comparalo con los datos de un certificado digital observado en alguna página de internet, por ejemplo la de un banco.

Ejercicio 4:

Alice quiere determinar un nivel de confianza para la firma de Fred.

La notación de cerificados usada es aumentada con H o con L para indicar si el que firma tiene un nivel mayor o menor de confianza en sus firmas.

Alice conoce y confía ampliamente en las opiniones de Harold y de Jane.
Alice apenas conoce a Tiago y por eso no sabe si sus opiniones son o no confiables.
A los demás participantes no los conoce.
Dadas las siguientes firmas, dar un argumento sólido, desde el punto de vista de Alice, por el cual la firma de Fred pueda ser confiable. X<<Y>> significa X certifica a Y
{Ellen(H), Tiago(H), George(H), Fred(H)}<<Fred>>
{Ellen(H), Harold(L), George(H)}<<George>>
{Jane(L), Harold(H),Ellen(H)}<<Ellen>>

Ejercicio 5:

El objetivo de este ejercicio es conocer la situación actual de infraestructura de firma digital en la República Argentina.

a) ¿Qué área del gobierno nacional actuará por ley como autoridad certificante raíz? (ACR RA)
b) Investiga cuáles son los certificadores licenciados vigentes del sistema de pki de la República Argentina. ¿Quién les otorgó la licencia?
c) según la ley 25506, ¿cuáles son las funciones de los certificadores licenciados?
d) ¿desde cuándo existe un certificado de la ACR RA? ¿para qué sirve?
e) Da un ejemplo de cadenas de firmas que podrían generarse a partir de la ACR RA.