GUÍA 5: MANEJO DE CLAVES – CIFRADO ASIMÉTRICO – FIRMA DIGITAL – SOLUCIONES

Ejercicio 1:

Ataques

1. **de replay.**

El intruso (o tramposo) repite o dilata información maliciosa o fraudulenta.

1) Alice envía a Bob un cheque digital firmado por ella para que él cobre $100.
2) Bob (tramposo) lo reenvía al banco más de una vez. El banco siempre certifica que está autorizado, y Bob le saca toda la plata a Alice.

Solución: timestamps.

2. **Key reuse.**

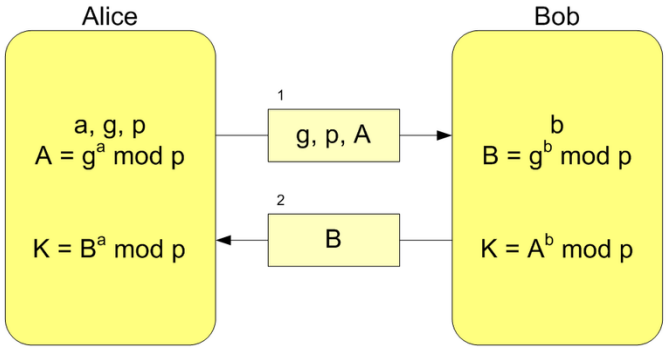
Como las claves de sesión se generan en forma seudoaleatoria, es posible predecir las claves de sesión y reusarlas.

1) En Needham Shroeder, si Mallory obtiene una clave de sesión anterior, y viene guardando los mensajes de Alice a Bob, puede enviarle a Bob un mensaje viejo $E_B(K,A)$. Bob extrae K y verifica que es de "Alice". Luego sigue el protocolo y genera un número aleatorio R_B , el cual encripta con la clave de sesión K. Envía eso a Alice: $E_K(R_B)$. Mallory intercepta este mensaje, y ella (en lugar de Alice) envía a Bob: $E_K(R_B-1)$. Bob se convence que se está comunicando con Alice, pero sigue la comunicación con Mallory.

Solución: timestamps. Buenos métodos de generación de claves aleatorias.

3. **Man in the middle**

El intruso actúa **sobre los dos canales** de comunicación (hacia A y hacia B). En general es también un "masquerading"



$K = A^b \bmod p = (g^a \bmod p)^b \bmod p = g^{ab} \bmod p = (g^b \bmod p)^a \bmod p = B^a \bmod p$

- Diffie Hellman.
- Alice envía el generador (g), el valor del número primo (p) y $g^a \bmod p$.
- Mallory intercepta el mensaje de Alice, y no puede hallar a . Pero puede, con el generador y el número primo generar un nuevo número $g^m \bmod p$ y enviarselo a Bob. Bob cree que es Alice, entonces sigue el protocolo enviándole $B=g^b \bmod p$, a la vez que calcula $K=(g^m)^b \bmod p$ como clave de sesión.
- Mallory, le hace algo parecido a Alice, por lo que queda generada una clave con Alice y otra con Bob. Luego Alice usará esa clave para encriptar sus mensajes a Bob (pero en realidad se estará comunicando con Mallory) y Bob usará otra clave para encriptar mensajes a Alice (pero en realidad sólo se comunica con Mallory que se entera de todo.)

4. **Masquerading**

Una persona o programa se hace pasar por otra falsificando datos. Ej. Diffie Hellman, Spoofing, suplantación de identidad.

Un man in the middle es, habitualmente un masquerading. Pero un masquerading no necesariamente es un man in the middle.

GUÍA 5: MANEJO DE CLAVES – CIFRADO ASIMÉTRICO – FIRMA DIGITAL – SOLUCIONES

Ejercicio 2:

Si Z intercepta los últimos mensajes, no puede desencriptar M porque debería tener la clave privada de B. Pero puede hacer un **man in the middle** de la siguiente manera:

1) Captura la clave pública de A y se la guarda.

$$A \rightarrow Z: \{ K_{UA} \}$$

2) Le pasa a B la clave pública suya.

$$Z \rightarrow B: \{ K_{UZ} \}$$

3) B le pasa a Z su clave pública, creyendo que se la pasa a A.

$$B \rightarrow Z: \{ K_{UB} \}$$

4) Z se queda con la K_{UB} y le envía a A la clave pública suya.

$$Z \rightarrow A: \{ K_{UZ} \}$$

5) Así, Z tiene el control de la comunicación, porque tanto A como B encriptarán sus mensajes con K_{UZ} y sólo Z los puede desencriptar con su clave privada y, si lo desea, reenviar al otro con modificaciones.

$$A \rightarrow B(Z): \{ E_Z(M)\}$$
$$A(Z) \rightarrow B: \{ E_B(M')\}$$
$$B \rightarrow A(Z): \{ E_Z(R)\}$$
$$B(Z) \rightarrow A: \{ E_A(R')\}$$

Ejercicio 3:

Cualquiera puede obtener, con la clave pública de A, el N1 y el Ks. Puede servir para garantizar identidad. Entonces habría que combinar encripción con firma:

1) Alice firma el mensaje con su clave privada y lo encripta con la clave pública de Bob:

$$A \rightarrow B: E_B\{S_A\{N1, K_s\}\}$$

2) Sólo Bob, con su privada, puede obtener $S_A\{N1, K_s\}$
3) Bob, con la clave pública de Alice obtiene $\{N1, K_s\}$. Sabe que sólo Alice lo pudo armar. El protocolo continúa igual.

$$B \rightarrow A: \{N1 + 1\}_{Ks}$$

Ejercicio 4:

A no debería permitir que Z envíe el mismo nonce, porque cualquiera se autenticaría correctamente reenviando el mismo mensaje.

Ejercicio 5:

Recordamos Needham y Schroeder:

1. **Alice → Trent: $\{A, B, rand_A\}$**

Alice inicia la comunicación con Trent.

2. **Trent →Alice: $\{rand_A, B, K_{Sesion}, \{A, K_{Sesion}\}_{KBT}\}_{KAT}$**

Trent genera una clave de sesión para las comunicaciones entre Alice y Bob.

Alice cuando recibe el mensaje, con K_{AT} efectúa la desencripción de lo que Trent le envió y obtiene:

$rand_A$ →que le confirma que este mensaje se corresponde con el 1.
B
K_{Sesion} -→ para sus futuros mensajes con Bob.
$\{A, K_{Sesion}\}_{KBT}$ -→ mensaje que le reenviará a Bob y que Alice no puede abrir.

3. **Alice → Bob : $\{A, K_{Sesion}\}_{KBT}$**

Bob recibe el mensaje de Alice, y con la clave K_{BT} (sólo conocida por él y Trent) lo abre obteniendo A y K_{sesion}.

4. **Bob →Alice: $\{ rand_B\}_{Ksesion}$**

Alice recibe el mensaje y lo puede abrir porque la clave de sesión es la que ella también obtuvo de Trent.

5. **Alice → Bob: $\{ rand_B - 1 \}_{Ksesion}$**

Bob recibe el mensaje y lo puede abrir y confirma $rand_B$.

Ataque:

Si Mallory consigue tener acceso a claves de sesión vieja (K_{sesion}), puede hacer lo siguiente:

1. Mallory(en nombre de Alice) → Bob: $\{ Alice, k_{session}\}_{KTB}$

Mallory le envió a Bob un mensaje viejo de los que interceptó entre Alice y Bob.

Bob lo recibe, y con la clave que comparte con Trent, lo abre y le envía a Mallory (creyendo que es Alice:

2. Bob → Mallory (en nombre de Alice):$\{rand_B\}K_{sesion}$

Como Mallory ve que puede desencriptar ese mensaje porque consiguió esa clave de sesión, entonces puede generar $rand_B$ -1 y engañar a Bob.

3. Mallory(en nombre de Alice) → Bob: $\{ rand_B - 1 \}_{Ksesion}$

Bob confirma $rand_B$ -1

Como Bob no inició el protocolo, no tiene manera de darse cuenta que el primer mensaje que Mallory le envió ($\{ Alice, k_{session}\}_{KTB}$) es en realidad un mensaje viejo. (Bob aparece en escena recién en el paso 3 del protocolo)

En esta nueva versión:

1. **No están hechos los pasos 1 y 2.**

Asumimos que Mallory pudo obtener, de intercambios anteriores, K_S y los mensajes intercambiados.

Entonces efectúa una repetición del mensaje 5:

Mallory(en nombre de Alice) → Bob: $\{ Alice, rand_X, k_{session}\}_{KTB}$

Cuando Bob recibe eso, si él mismo no envió recientemente el mensaje 2, (Bob→Alice: $\{ Alice, rand_X\}_{KBT}$)entonces se da cuenta que el mensaje que le acaba de llegar es falso. Lo rechaza.

Si él había enviado recientemente un mensaje 2, como el que le envía Mallory es anterior, seguramente **$rand_X$** y **$rand_X$** no coinciden, por lo tanto también lo rechaza.

2. **Están hechos los pasos 1 y 2.**

GUÍA 5: MANEJO DE CLAVES – CIFRADO ASIMÉTRICO – FIRMA DIGITAL – SOLUCIONES

Mallory envía su mensaje después de que Alice comenzó el protocolo.
Es decir:

1) A→B: A
2) B→A: $\{ A, rand_X \}_{KBT}$
3) A → Trent : { A, B, $rand_A$, $\{ A, rand_X \}_{KBT}$ }
4) Trent → A: { A, B, $rand_A$, $k_{session}$, $\{ A, rand_X, k_{session} \}_{KTB}$ }$_{KAT}$

Este mensaje, M no lo puede abrir porque no tiene k_{AT}.
Por lo tanto, a Bob le envía un mensaje viejo:

5) M(en nombre de A) → B: $\{ A, rand_X, k_{session} \}_{KTB}$

Bob abre el mensaje, y ve que $rand_x$ no coincide con $rand_x$, por lo tanto no continua el protocolo.

Otra opción es que Mallory empiece el protocolo como si fuera Alice.
Es decir:

1) Mallory (en nombre de A)→B: A
2) B→M (en nombre de A) : $\{ A, rand_X \}_{KBT}$
3) M saltea los pasos 3 y 4, y efectúa directamente el paso 5,

M(en nombre de A) → B: $\{ A, rand_X, k_{session} \}_{KTB}$

Pero como también lo tuvo que hacer con un mensaje viejo, nuevamente Bob abre el mensaje, y ve que $rand_x$ no coincide con $rand_x$.

Ejercicio 6: Group Diffie Hellman

Conocidos (g, p) los pasos serían::

1. $A \rightarrow B : g^x \bmod p = X$
2. $B \rightarrow C : g^y \bmod p = Y$
3. $C \rightarrow A : g^z \bmod p = Z$
4. $A \rightarrow B : \left(g^z\right)^x \bmod p = Z'$
5. $B \rightarrow C : \left(g^x\right)^y \bmod p = X'$
6. $C \rightarrow A : \left(g^y\right)^z \bmod p = Y'$

Después de esto, A, B y C tienen todos la clave k:

$\left(g^{yz}\right)^x \bmod p = k \quad = \quad \left(g^{xz}\right)^y \bmod p = k \quad = \quad \left(g^{xy}\right)^z \bmod p = k$

Ejercicio 7: Universidad de Saarland

a) Las firmas garantizan que fue Alice quien envió g^x y que fue Bob quien envió g^y.
b) Eso podría ocurrir si:

1. A→B(pero lo captura M): g^x
2. M → B: g^x
3. B→M: {B, certB, $S_B(g^x,g^y)$,g^y}
4. M(en nombre de B) →A: {B, certB, $S_B(g^x,g^y)$,g^y}
5. A→B(pero lo captura M): {A, certA, $S_A(g^x,g^y)$ }
6. M→B: {M, certM, $S_M(g^x,g^y)$ }

La firma que hace M es posible porque pudo ver en plano los valores g^x y g^y en los pasos 1 y 3.

Ejercicio 8:

MAC: es una función de hash de una sola via con el agregado de una clave secreta.
El que tiene la clave puede verificar el valor de hash.
Firma: es una encripción con una clave privada, que puede desencriptarse con una clave pública.

1. Puede Bob detectarlo con MAC o con firma. Es message integrity.
2. Es replay. Bob no puede detectarlo ni con MAC ni con firma.
3. Es un caso cheating. Se puede detectar con firma ya que sólo podrá desencriptarse con la pública del que lo hizo. Con MAC, tanto Oscar como Alice deberán revelar su clave privada para verificar el valor de hash y mostrar que es x.

GUÍA 5: MANEJO DE CLAVES – CIFRADO ASIMÉTRICO – FIRMA DIGITAL – SOLUCIONES

4. Es un caso de trampa (Bob is cheating). Con firma digital, Alice tiene que forzar a Bob a probar su reclamo enviándole una copia del mensaje y otra de la firma. Si Alice puede mostrar que se puede verificar con la clave pública de Bob, entonces fue Bob mismo el que envió el mensaje. Con MAC, no es posible verificar nada ya que Bob puede decir que él no sabe cómo se encriptó porque no tiene la clave con la que se hizo.

Ejercicio 9:

a) Tras el paso:

$\{E_B(S_A(K_s,time_A)), S_T(B, K_{UB}), S_T (A, K_{UA})\}$

Bob con su privada obtiene $S_A(K_s,time_A)$. Con la publica de Trent obtiene la clave pública de Alice que le permite verificar $S_A(K_s,time_A)$ y obtener la clave de sesión. A su vez, el valor timeA le sirve para detectar replay.

b) Bob puede hacer lo siguiente, una vez que completó el protocolo con Alice:

1. B → T: {B,C}
2. T → B: $\{S_T(B, K_{UB}), S_T (C, K_{UC})\}$
3. B(como alice)→C: $\{E_C(S_A(K_s,time_A)), S_T(C, K_{UC}), S_T (A, K_{UA})\}$

Carol se convence que está hablando con Alice.

Ejercicio 10: CERTIFICADOS DIGITALES

```
$ openssl genrsa -out priv.pem 1024
..............+++++
.....................................+++++
e is 65537 (0x010001)
$ openssl req -new -key priv.pem -out solicitud.csr
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]:AR
State or Province Name (full name) [Some-State]:CABA
Locality Name (eg, city) []:CABA
Organization Name (eg, company) [Internet Widgits Pty Ltd]:itbacys
Organizational Unit Name (eg, section) []:docencia
Common Name (e.g. server FQDN or YOUR name) []:AnaArias
Email Address []:mroig@itba.edu.ar

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

Ejercicio 11:

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

Las diferencias entre el archivo solicitud.csr (la solicitud de certificado) y autocertif.pem (el certificado en sí)

```
$ cat solicitud.csr
-----BEGIN CERTIFICATE REQUEST-----
MIIB8DCCAVkCAQAwgYUxCzAJBgNVBAYTAkFSMQ0wCwYDVQQIDARDQUJBMQ0wCwYD
VQQHDARDQUJBMRAwDgYDVQQKDAdpdGJhY3lzMREwDwYDVQQLDAhkb2NlbmNpYTER
MA8GA1UEAwwIQW5hQXJpYXMxIDAeBgkqhkiG9w0BCQEWEW1yb2lnQGl0YmEuZWR1
LmFyMIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQC93hmcVEo2IuH0GDjwh13Y
EQZVxpuA9v4YDQjclKFDY2TbFXHniDYBrz48XJ9IXNhbVfpWCnAc0UfHmPxDae/X
CMy1JO/J85lMcEkykvsMvA0P/AXiLMM7b29hLUlYLVMJ3407A7Pa46ibUJQr6yS6
14MXxBBik41ASJ+VaGDHFwIDAQABoCowEwYJKoZIhvcNAQkCMQYMBG90cmEwEwYJ
```

```
KoZIhvcNAQkHMQYMBGFiY2QwDQYJKoZIhvcNAQELBQADgYEAuZVXdejBN7HLkgTU
LPgIDoJNG21OMFOPhsqgFkSClEyGguLsV7n9Sl2bO4lAQJG/SKuv7oiFT4HD+e1d
KVYQmSNYUMPDuAF1vF0/iVzul/rArwHpRFi3lIzJuD1KwT/SIE1585zMlizfF81B
owfBBmG78OLZOUxqkok6t5u3A50=
-----END CERTIFICATE REQUEST-----
```

```
cat autocertif.pem
-----BEGIN CERTIFICATE-----
MIIC6DCCAlGgAwIBAgIUUvYVuKOrjHCaJqNI+wghyNpA47EwDQYJKoZIhvcNAQEL
BQAwgYUxCzAJBgNVBAYTAkFSMQ0wCwYDVQQIDARDQUJBMQ0wCwYDVQQHDARDQUJB
MRAwDgYDVQQKDAdpdGJhY3lzMREwDwYDVQQLDAhkb2NlbmNpYTERMA8GA1UEAwwI
QW5hQXJpYXMxIDAeBgkqhkiG9w0BCQEWEW1yb2lnQGl0YmEuZWR1LmFyMB4XDTIx
MDQyMDAwMDU0M1oXDTIxMDUyMDAwMDU0M1owgYUxCzAJBgNVBAYTAkFSMQ0wCwYD
VQQIDARDQUJBMQ0wCwYDVQQHDARDQUJBMRAwDgYDVQQKDAdpdGJhY3lzMREwDwYD
VQQLDAhkb2NlbmNpYTERMA8GA1UEAwwIQW5hQXJpYXMxIDAeBgkqhkiG9w0BCQEW
EW1yb2lnQGl0YmEuZWR1LmFyMIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQC9
3hmcVEo2IuH0GDjwh13YEQZVxpuA9v4YDQjclKFDY2TbFXHniDYBrz48XJ9IXNhb
VfpWCnAc0UfHmPxDae/XCMy1JO/J85lMcEkykvsMvA0P/AXiLMM7b29hLUlYLVMJ
3407A7Pa46ibUJQr6yS614MXxBBik41ASJ+VaGDHFwIDAQABo1MwUTAdBgNVHQ4E
FgQUhIsNTaM2OlZ1+YcZRDzZ98BGAK8wHwYDVR0jBBgwFoAUhIsNTaM2OlZ1+YcZ
RDzZ98BGAK8wDwYDVR0TAQH/BAUwAwEB/zANBgkqhkiG9w0BAQsFAAOBgQAyveNl
osxdsW4YJyJuA6pXiOZ1VHZsbBe24qJwztbTZpW/OE0oFFZyHO5Ygk9NDEdWeXuZ
oz45a1FSCDu+4gO0s/ocwlYX6znyrSdtkDjZm2BzsTk8OcsP9yEUo3/snsb6h+5G
1D/+ElguKjogU+UhEgWHMaPmgIj4+t3iSor5uA==
-----END CERTIFICATE-----
```

Ejercicio 12:

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**

```
$ openssl genrsa -out CApriv.key 1024
Generating RSA private key, 1024 bit long modulus (2 primes)
.......................+++++
....+++++
e is 65537 (0x010001)
$ openssl rsa -in CApriv.key -out CApub.key -pubout
writing RSA key
```

2) Luego, crea un archivo de texto llamado **CAconf1.cfg** con el siguiente contenido: (parámetros que se usarán para crear certificados digitales) y el archivo **CAconf2.cfg** (completa los campos de req_distinguished_name con los datos de quien será la autoridad certificante)

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) Crear un requerimiento de certificado con 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**

```
$ openssl genrsa -out USRpriv.key 1024
Generating RSA private key, 1024 bit long modulus (2 primes)
```

```
.....................................+++++
.............................................+++++
e is 65537 (0x010001)
$ openssl rsa -in USRpriv.key -out USRpub.key -pubout
writing RSA key
$ openssl req -new -key USRpriv.pem -out req.pem -config CAconf1.cfg
```

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

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.

```
$ openssl x509 -inform PEM -outform PEM -keyform PEM -CAform PEM -CAkeyform
PEM -in req.pem -out USRcert.cer -days 365 -req -CA ca.cer -CAkey CApriv.key
-sha1 -CAcreateserial -text
Signature ok
subject=C = AR, L = CABA, OU = docencia, CN = Anita, emailAddress =
mroig@itba.edu.ar
Getting CA Private Key
```

6) Observa el certificado obtenido (USRcert.cer).

```
$ cat USRcert.cer
Certificate:
    Data:
        Version: 1 (0x0)
        Serial Number:
            20:34:3b:bd:c9:2d:77:8d:8d:46:a0:07:5e:a4:fb:61:dc:7b:d8:a3
        Signature Algorithm: sha1WithRSAEncryption
        Issuer: C = AR
        Validity
            Not Before: Apr 20 00:56:06 2021 GMT
            Not After : Apr 20 00:56:06 2022 GMT
        Subject: C = AR, L = CABA, OU = docencia, CN = Ana, emailAddress =
mroig@itba.edu.ar
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                RSA Public-Key: (1024 bit)
                Modulus:
                    00:b3:62:50:da:ec:29:06:cb:48:e1:15:40:5c:2c:
                    14:9d:40:20:ea:fc:d4:eb:b1:7e:c7:e1:6a:3a:b6:
                    8b:99:42:70:00:31:18:4d:14:fa:e3:71:67:b8:b8:
                    fa:43:a7:32:e6:45:35:e4:9e:3b:d9:92:b9:f0:07:
                    8e:23:36:53:7e:69:a3:08:ed:30:68:54:60:af:d8:
                    a1:dd:02:15:90:8a:0e:19:b4:82:dd:bd:92:28:72:
                    27:60:c7:4a:88:33:f1:cf:f2:ab:49:2f:5c:d0:f9:
```

```
                    8d:58:03:b8:2a:8b:96:18:b5:cf:f8:2a:46:ac:95:
                    31:af:e8:c2:68:38:57:0a:7f
                Exponent: 65537 (0x10001)
    Signature Algorithm: sha1WithRSAEncryption
         89:ec:5a:18:82:e3:cd:c3:41:d9:f1:ae:70:66:b0:4a:1b:d0:
         b6:1f:60:21:41:52:62:80:7e:ed:f3:a4:06:d8:b3:37:77:bc:
         ad:c8:b4:67:ea:f2:4f:f1:98:43:03:34:95:d0:48:2a:53:d7:
         77:e4:11:37:89:4b:40:2b:d0:3c:01:c7:d0:bf:21:67:16:fe:
         d7:68:34:e0:3b:d4:33:ee:46:32:48:f5:0d:4b:3a:7f:3a:ad:
         f0:3c:dd:68:97:59:e0:06:0b:69:40:8f:4e:7f:47:90:da:1b:
         7a:9f:40:ba:56:f8:83:66:98:e4:b4:25:ed:8d:50:3a:a9:e9:
         a1:b5
-----BEGIN CERTIFICATE-----
MIIB8DCCAVkCFCA0O73JLXeNjUagB16k+2Hce9ijMA0GCSqGSIb3DQEBBQUAMA0x
CzAJBgNVBAYTAkFSMB4XDTIxMDQyMDAwNTYwNloXDTIyMDQyMDAwNTYwNlowYTEL
MAkGA1UEBhMCQVIxDTALBgNVBAcMBENBQkExETAPBgNVBAsMCGRvY2VuY2lhMQ4w
DAYDVQQDDAVBbml0YTEgMB4GCSqGSIb3DQEJARYRbXJvaWdAaXRiYS5lZHUuYXIw
gZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBALNiUNrsKQbLSOEVQFwsFJ1AIOr8
1OuxfsfhajQ2i5lCcAAxGE0U+uNxZ7i4+kOnMuZFNeSeO9mSufAHjiM2U35powjt
MGhUYK/Yod0CFZCKDhm0gt29kihyJ2DHSogz8c/yq0kvXND5jVgDuCqLlhi1z/gq
RqyVMa/owmg4Vwp/AgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAiexaGILjzcNB2fGu
cGawShvQth9gIUFSYoB+7fOkBtizN3e8rci0Z+ryT/GYQwM0ldBIKlPXd+QRN4lL
QCvQPAHH0L8hZxb+12g04DvUM+5GMkj1DUs6fzqt8DzdaJdZ4AYLaUCPTn9HkNob
ep9Aulb4g2aY5LQl7Y1QOqnpobU=
-----END CERTIFICATE-----
```

Ejercicio 13:

Alice confía en Harold.

Harold firma el certificado de Ellen con un alto nivel de confianza.

Ellen firma el ceritificado de Fred con un alto nivel de confianza.

Entonces Alice puede dar al certificado de Fred un alto nivel de confianza.

Ejercicio 14: https://pki.jgm.gov.ar/app/ **y** https://www.acraiz.gob.ar/

a) ¿Qué área del gobierno nacional actuará por ley como autoridad certificante raíz? (ACR RA)

https://www.acraiz.gob.ar/Home/Normativa

La Oficina Nacional de Tecnologías de Información, (ONTI) deberá actuar como Autoridad Certificante.
El Decreto Nº 409/2005 asignó a la Subsecretaría de la Gestión Pública de la Jefatura de Gabinete de Ministros las funciones de entidad licenciante de certificadores. (Articulo 2, inciso 16: " Actuar como autoridad de aplicación del Régimen Normativo que establece la infraestructura de Firma Digital establecida en la Ley N° 25.506, como así también en las funciones de entidad licenciante de certificadores, supervisando su accionar.").

GUÍA 5: MANEJO DE CLAVES – CIFRADO ASIMÉTRICO – FIRMA DIGITAL – SOLUCIONES

En 2016, la Resolución 399e/2016, modifica algunas cuestiones dada la nueva organización de Ministerios
En su artículo 14 establece: "**La Autoridad Certificante Raíz es la Autoridad Certificante administrada por la SECRETARÍA DE MODERNIZACIÓN ADMINISTRATIVA del MINISTERIO DE MODERNIZACIÓN. Constituye la única instalación de su tipo y reviste la mayor jerarquía de la Infraestructura de Firma Digital de la REPÚBLICA ARGENTINA. Emite certificados digitales a las Autoridades Certificantes de los certificadores licenciados, una vez aprobados los requisitos de licenciamiento**."

b) Investiga cuáles son los certificadores licenciados vigentes del sistema de pki de la República Argentina. ¿Quién les otorgó la licencia?

La lista de Certificadores Licenciados vigentes (Ley 25.506 art. 17) estaba conformada, en los comienzos de la aplicación de la Ley por:

5. AFIP - Administración Federal de Ingresos Públicos
6. ANSeS - Administración Nacional de la Seguridad Social

Al día de hoy, según https://www.acraiz.gob.ar/Home/CertificadoresLicenciados

1. AFIP – Administración Federal de Ingresos Públicos
2. ANSeS – Administración Nacional de la Seguridad Social
3. ONTI – Oficina Nacional de Tecnologías de Información
4. Encode SA (entidad privada)
5. LAKAUT S.A. (entidad privada)
6. BOX CUSTODIA DE ARCHIVOS S.A. (entidad privada)
7. DIGILOGIX S.A. (entidad privada)
8. TRAIN SOLUTIONS S.A. (entidad privada)
9. TECNOLOGIA DE VALORES S.A. (entidad privada)
10. PRISMA MEDIOS DE PAGO S.A. (entidad privada)

ARTICULO 17. — Del certificador licenciado. Se entiende por certificador licenciado a toda persona de existencia ideal, registro público de contratos u organismo público que expide certificados, presta otros servicios en relación con la firma digital y cuenta con una licencia para ello, otorgada por el ente licenciante. *(ente licenciante es el Ministerio de Modernización)*
La actividad de los certificadores licenciados no pertenecientes al sector público se prestará en régimen de competencia. El arancel de los servicios prestados por los certificadores licenciados será establecido libremente por éstos.

c) según la ley 25506, ¿cuáles son las funciones de los certificadores licenciados?

Los **certificadores licenciados** son entidades públicas o privadas que se encuentran habilitados por el Ente Licenciante para emitir certificados digitales, en el marco de la Ley 25.506 de Firma Digital.

ARTICULO 19. — Funciones. El certificador licenciado tiene las siguientes funciones:
a) Recibir una solicitud de emisión de certificado digital, firmada digitalmente con los correspondientes datos de verificación de firma digital del solicitante;
b) Emitir certificados digitales de acuerdo a lo establecido en sus políticas de certificación, y a las condiciones que la autoridad de aplicación indique en la reglamentación de la presente ley;
c) Identificar inequívocamente los certificados digitales emitidos;
d) Mantener copia de todos los certificados digitales emitidos, consignando su fecha de emisión y de vencimiento si correspondiere, y de sus correspondientes solicitudes de emisión;
e) Revocar los certificados digitales por él emitidos en los siguientes casos, entre otros que serán determinados por la reglamentación:

1) A solicitud del titular del certificado digital.
2) Si determinara que un certificado digital fue emitido en base a una información falsa, que en el momento de la emisión hubiera sido objeto de verificación.
3) Si determinara que los procedimientos de emisión y/o verificación han dejado de ser seguros.
4) Por condiciones especiales definidas en su política de certificación.
5) Por resolución judicial o de la autoridad de aplicación.

f) Informar públicamente el estado de los certificados digitales por él emitidos. Los certificados digitales revocados deben ser incluidos en una lista de certificados revocados indicando fecha y hora de la revocación. La validez y autoría de dicha lista de certificados revocados deben ser garantizadas.

d) ¿desde cuándo existe un certificado de la ACR RA? ¿para qué sirve?

Desde 2007, se usó el AC Raiz-RA 2007 (ca.crt)

Desde 2016, AC Raiz RA V2.0 (acraizra.crt)

La autoridad Certificante Raiz:

- Emite, renueva y revoca su propio certificado digital.
- *Emite, renueva y revoca los certificados de las autoridades certificantes de los certificadores licenciados.*
- Emite su LCR

https://www.acraiz.gob.ar/Content/Archivos/Normativa/2016-37e%20(Res.%20SMA)%20-%20AC%20RAIZ%20v2.0%20(ANEXO%20I).pdf