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

En los ejercicios de esta guía, considerar:

- K_{sx} es clave privada de x; K_{px} es clave pública de x;
- E_x es encripción con clave pública de x;
- D_x es desencripción con clave privada de x;
- S_x es firma con clave privada de x;
- V_x es verificación con clave pública de x;
- K_{xy} o K_s es clave de sesión compartida entre x e y;
- $\{M\}_{KS}$ es encripción de M con clave simétrica KS
- Mallory efectúa ataques activos y Eve efectúa ataques pasivos.

Ejercicio 1:

Escribe un ejemplo de los siguientes ataques que pueden darse contra un protocolo. ¿Pueden evitarse?

1) Replay
2) Key Reuse
3) Man in the middle
4) Masquerading (suplantación de identidades)

Ejercicio 2:

Considera el siguiente protocolo para enviar un texto plano M entre A y B:

1) A → B: $\{ K_{pA} \}$
2) B → A: $\{ K_{pB} \}$
3) A → B: $\{ E_B(M) \}$
4) B → A: $\{ E_A(M) \}$

Si un adversario (Z) intercepta el primer mensaje, ¿cómo hace para obtener el texto plano M?

Ejercicio 3:

¿Cuál es el problema con el siguiente protocolo? Solucionarlo.

1) A → B: $S_A\{N1, K_s\}$
2) B → A: $\{N1 + 1\}_{Ks}$

Ejercicio 4:

Considera el siguiente protocolo de autenticación mutua en el cual A y B se autentican mutuamente intercambiando 4 mensajes:

1) A → B: N1
2) B → A: N2
3) A → B: $(N2)_{Ks}$
4) B → A: $(N1)_{Ks}$

Donde:

- N1 y N2 son números generados en forma aleatoria (nonce)

- A y B son los ID de las partes intervinientes

- Ks es una clave simétrica ya compartida entre A y B

A autentica con éxito a B al recibir el cuarto mensaje y B autentica con éxito a A al recibir el tercer mensaje. Como Ks es una clave ya compartida entre A y B solamente, cualquiera que encripte un mensaje usando Ks se asegura que posee K y por lo tanto queda autenticado.

¿Qué situación *NO* debe permitir A para evitar que un tercero no autorizado se autentique correctamente?

Ejercicio 5:

En el protocolo original de Needham Schroeder, cuando se roban claves de sesión es posible un ataque de replay.

La siguiente es una variante del protocolo de Needham Schroeder:

1. Alice → Bob: Alice
2. Bob →Alice: $\{ \text{Alice}, \text{rand}_X \}_{KBT}$
3. Alice → Trent: $\{ \text{Alice}, \text{Bob}, \text{rand}_A, \{ \text{Alice}, \text{rand}_X \}_{KBT} \}$
4. Trent → Alice: $\{ \text{Alice}, \text{Bob}, \text{rand}_A, k_{session}, \{ \text{Alice}, \text{rand}_X, k_{session} \}_{KTB} \}_{KAT}$
5. Alice → Bob: $\{ \text{Alice}, \text{rand}_X, k_{session} \}_{KTB}$
6. Bob →Alice: $\{ \text{rand}_B \}_{Ks}$
7. Alice → Bob: $\{ \text{rand}_B - 1 \}_{Ks}$

Mostrar que con esta variante se resuelve el problema de ataque de repetición.

Ejercicio 6:

Considera el protocolo de intercambio de claves Diffie Hellman y escribe la secuencia de pasos para que en lugar de ser 2 los participantes que generan una clave compartida sean 3.

Ejercicio 7:

Considera un protocolo normal de intercambio de claves Diffie – Hellman con autenticación. El objetivo es proveer autenticación mutua con intercambio de claves. Asumimos que cada parte tiene una clave privada para firmar en algún esquema de firma y un certificado con la correspondiente clave pública. El protocolo procede de la siguiente manera:

1) A→B: g^x
2) B→A: $\{B, certB, S_B(g^x,g^y), g^y\}$
3) A→B: $\{A, certA, S_A(g^x,g^y) \}$

Finalmente, Alice y Bob pueden calcular la clave compartida y secreta $K = g^{xy}$.

a) Explicar el por qué de las firmas en el protocolo anterior.

b) Mostrar que un atacante activo, Mallory, puede interferir con el protocolo mediante un ataque **man in the middle** tal que al final tendremos la siguiente situación:

- Alice cree que se está comunicando de forma segura con Bob
- Pero Bob cree que se está comunicando de forma segura con Mallory

Ejercicio 8:

En este problema se comparan los servicios que provee la firma digital y los códigos de autenticación de mensajes (MAC).

Se asume que Oscar puede observar los mensajes que Alice y Bob se envían, pero no conoce ninguna clave, salvo las públicas.

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

Determinar si el ataque se puede detectar o proteger con la Firma Digital, con el código de autenticación MAC, con ambos o con ninguno. Clasificar el tipo de ataque (man in the middle, replay, message integrity, cheating, etc.)

a) Alice envía un mensaje *x = "Trasferir $1000 a Mark"* en plano y también envía *sign(x)* a Bob. Oscar intercepta el mensaje y reemplaza "*Mark"* con "*Oscar"*. ¿Puede Bob detectar esto?

b) Alice envía un mensaje *x = "Trasferir $1000 a Oscar"* en plano y también envía *sign(x)* a Bob. Oscar observa el mensaje y la firma y lo reenvía 100 veces a Bob. ¿Puede Bob detectar esto?

c) Oscar afirma que él envió un mensaje x con firma válida *sign(x)* a Bob. Alice afirma que fue ella. ¿Puede Bob dirimir la cuestión?

d) Bob dice que recibió un mensaje *x = "Trasferir $1000 de Alice a Bob"* con firma válida *sign(x)* de parte de Alice. Pero Alice dice que ella nunca mandó eso. ¿Puede Alice aclarar su situación?

Ejercicio 9:

El siguiente protocolo usa criptografía de clave pública. Trent tiene una base de datos con todas las claves públicas de los participantes.

1) A→T: {A, B}
2) T→A: $\{S_T(B, K_{pB}), S_T(A, K_{pA})\}$
3) A→B: $\{E_B(S_A(K_s, time_A)), S_T(B, K_{pB}), S_T(A, K_{pA})\}$

a) Explicar qué hace Bob después del paso 3 para ratificar que puede comunicarse con Alice con seguridad.

b) Explicar cómo hace Bob para impersonarse como Alice frente a Carol (masquerading)

Ejercicio 10: 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 11:

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 12:

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

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 13:

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 14:

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.