Universidad Politécnica de Cartagena

Escuela Técnica Superior de Ingeniería de Telecomunicación

SEGURIDAD EN REDES

Práctica 3: Usos cotidianos de la criptografía (III):

Firma electrónica
Seguridad en el correo electrónico

Autores: Josemaría Malgosa Sanahuja
María Dolores Cano Baños

SEGURIDAD EN REDES

Usos cotidianos de la criptografía PRÁCTICA 3 (tercera parte)

FIMA ELECTRÓNICA

La firma electrónica es un procedimiento digital que pretende sustituir a la firma manuscrita de un documento impreso. Aunque los protocolos de firma electrónica son conocidos desde hace tiempo, también hay que reconocer que todos ellos se desarrollaron a mediados del siglo XX. Son por tanto, jóvenes. Precisamente por ello, las empresas y las instituciones públicas han tardado en asumir dicha tecnología para sus quehaceres cotidianos. Lo que significa que, aunque ya existe legislación regulatoria para la firma electrónica (Ley 59/2003) y estándares consolidados para realizarla, sigue siendo un mundo en evolución.

Existen dos tipos de firma digital: la simple y la avanzada. La firma digital simple permite: (1) identificar al firmante y (2) detectar cualquier cambio ulterior de los datos firmados. Aunque existen muchos protocolos para la firma electrónica, el más utilizado es el basado en claves asimétricas RSA.

1. Describir sucintamente- pero con detalle- en qué consiste la firma electrónica RSA

2. Crear el archivo *tesisidoctoral.txt* que contenga el siguiente texto: "*Con un 6 y con un 4 aquí tiene su retrato*". Utilizando *openssl* firmar electrónicamente el archivo *tesisdoctoral.txt* (llamar a la firma *tesisdoctoral.sig*). ¿Qué tamaño tiene la firma?

```
alumno@localhost:~/pr3> openssl dgst -sha1 -sign FNMTpriv.pem  -out tesisdoctoral.signed tesisdoctoral.txt
```

FNMTpriv: 2.5 KiB
Archivo: 128 bytes

3. Transferir los dos archivos al usuario remoto y que éste se encargue de verificar la firma con *openssl*. Verificar la robustez de la firma RSA frente a suplantación de identidad o error de integridad

```
ser70@labit201:~/pr3> openssl dgst -sha1 -verify FNMTpum.pem -signature tesisdoctoral.signed tesisdoctoral.txt
```

Los países miembros del Espacio Europeo han estandarizado tres formatos para la firma digital avanzada: CAdES, XAdES y PAdES (denominados de forma genérica AdES). Además, también existen formatos de firma avanzada para correo electrónico (openPGP), documentos Microsoft Office (OOXML) y libreoffice (ODF). La firma digital avanzada permite:

- Identificar al firmante y detectar cualquier cambio ulterior de los datos firmados
- Añadir marcas (o sellos) temporales:

Para verificar una firma es necesario comprobar la integridad de los datos firmados asegurando que éstos no hayan sufrido ninguna modificación y comprobar que el estado del certificado con el que se firmó era el correcto, es decir, era vigente en el momento de la operación.

En el caso de la firma electrónica básica, si el certificado está caducado, automáticamente se da la firma como no válida. Entonces, ¿cómo sabemos que el certificado estaba vigente o no en la fecha en la que se firmó? ¿Qué debe hacerse para que cuando se quiera validar o verificar una firma en el futuro, la validación sea posible aunque esté caducado el certificado?

Para dar respuesta a estas preguntas, la firma avanzada contempla la posibilidad de incorporar a las firmas información adicional que garantiza la validez de una firma a largo plazo una vez vencido el periodo de validez del certificado. Estos formatos añaden a la firma evidencias de terceros (de autoridades de certificación) y sellos de tiempo, que realmente certifican cuál era el estado del certificado en el momento de la firma. Cuando el sello temporal no se ha emitido por una autoridad de certificación externa al proceso de firmado, se le denomina marca de tiempo

- Incluir en la firma el archivo original
- La firma múltiple: un documento pueda ser firmado por varias personas
  - Co-firma o firma en línea. Todos los firmantes están al mismo nivel y en la que no importa el orden en el que se firma
  - Contra-firma o firma en cascada. El orden en el que se firma es importante, ya que cada firma debe refrendar o certificar la firma del firmante anterior.
- Incluir los certificados de los firmantes o al menos, referencias de verificación de los certificados (a través de accesos a CRL)

4. Describir sucintamente- pero con detalle- en qué consisten los estándares CAdES (y su versiones T, C, X, XL y A), XAdES, PAdES, OOXML y ODF

5. Firmar el documento Rijndael.pdf siguiendo la especificación PAdES con la aplicación @Firma del Ministerio de Hacienda y Administraciones Públicas (autofirma)

6. Verificar el proceso de firma con la propia aplicación

7. ¿En qué consiste la firma múltiple?

   Varios usuarios pueden firmar el mismo archivo

SEGURIDAD EN EL CORREO ELECTRÓNICO (openPGP)

8. Describir sucintamente- pero con detalle- en qué consiste openPGP

Abrir *thunderbird* y configurarlo para que pueda leer y enviar correos electrónicos utilizando tu cuenta de correo electrónico (personal y la de la upct)

9. Enigmail es una implementación de OpenPGP para *thunderbird*. Habilitar Enigmail y crear las claves pública y privada de ambas cuentas

10. Ralizar las siguientes acciones: (1) Envíar un correo electrónico firmado de una cuenta a otra y que el destinatario verifique la firma a (2) Enviar un correo cifrado y firmado de una cuenta a otra y que el destinatario verifique la firma

En realidad, Enigmail no es más que un *front-end* de *gpg2* (GNU Privacy Guard). Por lo tanto, todo lo que hace Enigmail se hace en realidad con comandos *gpg2*. En el portal de la asignatura encontrará una guía completa de cómo utilizar *gpg2*

11. Salir del *thunderbird* y eliminar los directorios ~/.thunderbird y ~/.gnupg. Crear el archivo *msg.txt* y escribir en él el siguiente texto "*Este mensaje es secreto: sin pilas, los mandos a distancia dejan de funcionar*". Abrir una consola y ejecutar (y explicar el funcionamiento) de los siguientes comandos:

    Genera certificado del usuario -> Equivalente a generarlo desde el enigmail

    `gpg --gen-key`
    Identificar las claves generadas con el nombre *alumno* y el mail *alumno@upct.es*. Como *passphrase* poner *alm*

    `gpg --list-keys` muestra las claves que están almacenadas
    `gpg --list-secret-keys`

    `gpg --import JMS_pub.pgp`[1]

```
alumno@localhost:~/pr3> gpg --import JMS_pub.pgp
gpg: key F9DD9890D6761A4D: public key "Josemaria Malgosa Sanahuja <josem.malgosa@upct.es>" imported
gpg: Total number processed: 1
gpg:               imported: 1
```

    Verificar que la clave pública del profesor de la asignatura se ha importado correctamente ¿Cuál es su identificador?

    `gpg --delete-key josem.malgosa@upct.es`
    Verificar que la clave pública del profesor de la asignatura se ha eliminado

[1] JMS_pub.pgp es la clave PGP pública del profesor y está disponible en el portal de la asignatura

```
gpg -a --output pub.pgp --export alumno@upct.es
gpg -a --output priv.pgp --export-secret-key alumno@upct.es
gpg --import SER_pub.pgp[2]
```
Verificar que la clave pública del usuario remoto se ha importado correctamente ¿Cuál es su identificador?

```
gpg -a --output msg.pgp --encrypt --recipient <ID usuario remoto> msg.txt
gpg -a --output msg.sig.pgp --sign msg.txt
gpg --clearsign msg.txt
```

12. Mandar el mensaje *msg.pgp* al usuario remoto y que éste descifre el mensaje (la *passphrase* de la clave privada del usuario remoto es `lsnei`)

13. Mandar el mensaje *msg.sig.pgp* al usuario remoto y que éste: (1) verifique solo la firma y (2) simultáneamente descifre el mensaje y verifique la firma.

    El usuario tiene que tener la clave pública de la persona que firma el mensaje

> Todo esto es lo que Enigmail hace sin que el usuario se dé cuenta. Si no se dispone de un *front-end* del tipo Enigmail, habrá que ejecutar primero estos comandos a mano para después enviar los ficheros cifrados y firmados mediante un cliente de correo tradicional. En cualquier caso, openPGP puede ser utilizado para firmar y cifrar documentos que nada tienen que ver con el correo electrónico, de la misma forma que lo hace *openssl*, (aunque *openssl* tiene muchas otras funcionalidades). Existen *front-ends* para este propósito como por ejemplo *Kleopatra* en entornos KDE.

14. ¿Para qué sirve firmar con la opción `-detach-sig`? En este caso, ¿cómo se comprobaría la validez de la firma? Para verificar que el correo/fichero no ha sido manipulado

15. Descargarse del portal de la asignatura el archivo Actas_Jitel_2009.pdf y verificar su integridad (es decir, que durante la descarga el documento no ha sufrido ninguna alteración):

    - Calculando su hash MD5 y verificar que coincide con el que aparece en el portal
    - Calculando su hash SHA-1 y verificar que coincide con el que aparece en el portal

16. La firma digital también puede utilizarse para verificar la integridad de un documento. Verificar la firma digital *Actas_Jitel_2009signed.pgp*; es decir, que el archivo *Actas_Jitel_2009.pdf* ha sido firmado por el profesor de la asignatura. Verificar también que el *fingerprint* de la clave pública del profesor es el que aparece en el portal.

```
alumno@localhost:~/pr3> md5sum Actas_Jitel_2009.pdf
3836db5b5ce70c8b724ef9d20f9a31e1  Actas_Jitel_2009.pdf
alumno@localhost:~/pr3> sha1sum Actas_Jitel_2009.pdf
2af93f0b396d3993b547b083863f7dfb467673fc  Actas_Jitel_2009.pdf
```

```
alumno@localhost:~/pr3> gpg --verify Actas_Jitel_2009signed.pgp Actas_Jitel_2009.pdf
```

[2] SER_pub.pgp es la clave PGP del usuario remoto y está disponible en el portal de la asignatura