Seguridad en Redes

Práctica 2.1. Criptografía de clave secreta y funciones resumen

OpenSSL

OpenSSL es una herramienta de software libre muy potente que incluye bibliotecas criptográficas. Estas bibliotecas suministran funciones criptográficas a otros paquetes como OpenSSH y ayudan a implementar protocolos y algoritmos de SSL (*Secure Socket Layer*) y TLS (*Transport Layer Security*). Ambos protocolos relacionados con la seguridad en navegadores web (protocolo *https*).
OpenSSL es válido para cualquier sistema operativo (Linux, Unix, Windows, etc). En la página web del proyecto (http://www.openssl.org) podemos encontrar información exhaustiva sobre el uso de esta herramienta.

- Revisa el manual de los comandos `openssl` que encontrarás en esta página (https://www.openssl.org/docs/man1.0.2/apps/).

Cifrado de bloque

Para cifrar un fichero con un algoritmo de cifrado simétrico, se usa el comando `enc` con la opción `-e` (usada por defecto), indicando un algoritmo de cifrado de bloque (`aes-128-ecb`, `aes-128-cbc`...):

```
$ openssl enc [-algoritmo] [-e] -in plain.txt -out cipher.bin
```

donde `plain.txt` es el fichero que contiene el texto en claro y `cipher.bin` es el fichero que contiene el texto cifrado.

También se puede usar:

```
$ openssl [algoritmo] [-e] -in plain.txt -out cipher.bin
```

Para descifrarlo, también se usa el comando `enc` con la opción `-d`:

```
$ openssl enc [-algoritmo] -d -in cipher.bin -out plain-again.txt
```

También se puede usar:

```
$ openssl [algoritmo] [-d] -in cipher.bin  -out plain.txt
```

Los algoritmos soportados se pueden ver con `openssl enc -help` o en la página de manual de `enc`.

- Cifra y descifra ficheros con distintos algoritmos.

- Cifra un fichero varias veces con el mismo algoritmo y con la misma contraseña. Compara los textos cifrados con `cmp` y observa su contenido con `xxd`. ¿Son iguales? ¿Por qué? ¿Cómo genera, por defecto, Openssl la clave de cifrado? Si hacemos lo mismo usando la opción –nosalt, ¿qué ocurre? ¿por qué?

    - **Entrega:** Contesta a las preguntas formuladas justificando las respuestas adecuadamente.

- Consulta las opciones `-p` y `-P`, `-pass`, `-S` y `-K` y `-iv` en el manual y úsalas tanto para cifrar como para descifrar.

    - **Entrega:** Explica para qué sirve cada una de las opciones probadas. Indicar, también, cómo consigue el receptor el IV y la salt para poder descifrar el mensaje.

Modos de bloque

- Cifra la imagen `tux.bmp` (disponible en el Campus Virtual) usando los modos de bloque ECB y CBC. Guarda el resultado en los ficheros `tux-ecb.bmp` y `tux-cbc.bmp`, respectivamente.

  Para poder ver la imagen cifrada, es necesario restaurar la cabecera del formato BMP (primeros 54 bytes). Para ello, copia la cabecera del fichero original al fichero cifrado con:

```
$ dd if=tux.bmp of=tux-ecb.bmp bs=1 count=54 conv=notrunc
$ dd if=tux.bmp of=tux-cbc.bmp bs=1 count=54 conv=notrunc
```

- Observa las diferencias en las imágenes y prueba otros modos de bloque.

    - **Entrega:** Copia los ficheros `tux-ecb.bmp` y `tux-cbc.bmp`. Explica cómo se ven estos ficheros, qué diferencia hay entre ellos, y relaciona todo esto con el funcionamiento del modo de bloque con el que se ha cifrado.

Cifrado de flujo

Para cifrar, se usa el comando `enc` con un algoritmo de flujo (por ejemplo, `rc4`).

- Prueba el programa `rc4.c` (disponible en el Campus Virtual) con el algoritmo RC4 y compáralo con el comando equivalente de `openssl` (ver trasparencia donde se encuentra el código RC4, ejecutar el contenido del cuadro inferior de esa transparencia).

    - **Entrega:** ¿cuál es el texto en claro? ¿cuál es la clave? ¿cuál es el valor de la clave que se introduce cuando se ejecuta el comando `openssl`?

Funciones *hash* y HMAC

Para obtener un resumen (*digest*) de un fichero, se usa el comando `dgst` con un algoritmo (`md5`, `sha1`, `sha256`…):

```
$ openssl dgst [-algoritmo] fichero
```

También se puede usar:

```
$ openssl [algoritmo] fichero
```

Los algoritmos soportados se pueden ver con la opción `-help` de `dgst` o con `man dgst`.

- Calcula códigos *hash* de ficheros con distintos algoritmos.

  - **Entrega:** Copia diferentes códigos *hash* del fichero `/etc/services`. Especificar el algoritmo de hash usado para cada función resumen y su tamaño en bits.

La opción `-hmac` permite obtener un HMAC (*Keyed-Hash Message Authentication Code*) tal y como como se usa en RIP o en IPsec:

```
$ openssl dgst [-algoritmo] -hmac <key> fichero
```

- Calcula códigos HMAC de ficheros con distintos algoritmos.

  - **Entrega:** Copia diferentes códigos HMAC del fichero `/etc/services` con la contraseña “seguridad”. ¿De qué depende el valor del código HMAC de un fichero?

GnuPG

GnuPG es la implementación completa y libre del estándar OpenPGP [RFC 4880] del proyecto GNU, antes conocido como PGP (Pretty Good Privacy). Permite cifrar y firmar datos, proporciona un sistema de gestión de claves versátil y módulos de acceso a todo tipo de directorios de claves públicas. Además, proporciona un conjunto de herramientas de línea de comandos que puede ser integrada en otras herramientas. La herramienta gpg2 implementa la parte OpenPGP, proporcionando servicios de cifrado y firma digital.

- Revisa la página de manual de `gpg2`.

Cifrado y descifrado

Para cifrar con clave simétrica, se usa la opción `--symmetric`:

```
$ gpg2 --symmetric [--cipher-algo <algoritmo>] [-a] plain.txt
```

La opción `-a` (o `--armor`) escribe el resultado en ASCII (usando Base64) en lugar de hacerlo en binario.

Los algoritmos soportados se pueden ver con la opción `--version`:

```
$ gpg2 --version
```

Para descifrar, se usa simplemente:

```
$ gpg2 plain.gpg
```

O, si se usó la opción `-a`:

```
$ gpg2 plain.asc
```

- Cifra y descifra ficheros con distintos algoritmos.
    - **Entrega:** Copia el resultado de cifrar el fichero `/etc/services` (copiándolo primero al directorio `$HOME`) con la contraseña “seguridad”. Especificar el algoritmo de cifrado usado, así como el tamaño de la clave y el modo de operación.

Funciones *hash*

Se usa el comando `--print-md`:

```
$ gpg2 --print-md <algoritmo> plain.txt
```

También se puede usar:

```
$ gpg2 --print-mds plain.txt
```

- Calcula códigos *hash* de ficheros con distintos algoritmos.
    - **Entrega:** Copia varios códigos *hash* del fichero `/etc/services`. Especificar el algoritmo de hash usado para cada función resumen y su tamaño en bits.