Seguridad en Redes y Telecomunicaciones

Grado en Criminología
Curso 2017-18

Guía de Prácticas

Práctica 2.- Identidad digital y privacidad
Sesión 3. Cifrado de datos

Objetivo: Hoy trabajaremos con dos herramientas que nos permiten mejorar la privacidad:
- Cifrado
- Esteganografía.
- Generación de hashes.

1.- ¿Qué es el cifrado o encriptación?

Vamos a introducir los principios básicos de la criptografía para comprender las bases de las herramientas que vamos a utilizar en la práctica.

Entendemos por encriptación (cifrado) la conversión de datos de un formato a otro no legible. Esta transformación ayuda a proteger la privacidad de la información tanto de los archivos almacenados en nuestro sistema o como de los mensajes enviados a un receptor. Una vez realizado el cifrado, para poder leer la información almacenada o enviada, debemos deshacer el proceso, desencriptar (descifrar) los datos, es decir, devolverlos a su formato original.

El proceso de encriptación/desencriptación necesita información adicional, esta información es lo que denominamos *clave*. En algunos casos se utiliza la misma clave para los dos pasos, en otros, se utiliza una clave diferente para cada uno.

Existen tres tipos de técnicas criptográficas:

- **Criptografía de clave secreta**: se utiliza una única clave tanto en encriptación como en la desencriptación, por ello hablamos de *encriptación simétrica* (ver Figura 1).

- **Criptografía de clave publica**: se utilizan dos claves para asegurar las comunicaciones entre emisor-receptor cuando la comunicación se realiza por canales inseguros. Al usar dos claves, hablamos de *encriptación asimétrica*. En este sistema, cada parte tiene dos claves, una clave pública una privada. La privada es un secreto, mientras que la pública es compartida por todos aquellos con los que nos queremos comunicar. Cuando A quiere enviar un mensaje a B, A utiliza la clave pública de B para cifrarlo, y solo B puede descifrarlo utilizando su clave privada (Figura 1).

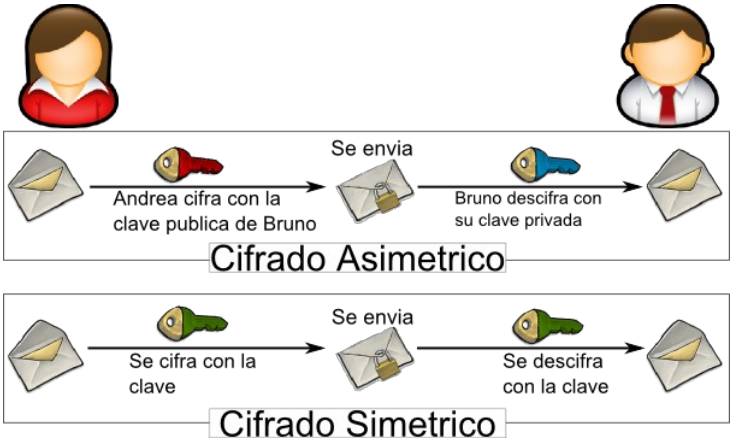

Figura 1.- Cifrado simétrico y asimétrico.

- **Funciones hash**: estas funciones no involucran ninguna clave, en su lugar, producen un valor hash (resumen) de longitud fija que se calcula en base a un mensaje de texto plano. Las funciones hash se utilizan para comprobar la integridad o autenticación de los mensajes y asegurarse de que no han sido alterados, comprometidos o infectados por un virus. Algunos algoritmos hash son: MD-5, SHA-1, SHA-2 y SHA-3.

En esta práctica vamos a ver como se encripta/cifra una archivo y un sistema de archivos con el primer método, el de clave secreta.

2.- Cifrado de archivos

Para el cifrado de archivos, vamos a utilizar una herramienta denominada **AESCrypt**[1]. Para su instalación, una vez descargado el archivo `AESCrypt_v310_win32.zip`, debemos descomprimirlo y ejecutar el programa `setup.exe`.

Una vez instalado, para cifrar un archivo solo debemos seguir los pasos:

1. Pulsar el botón derecho del ratón sobre el archivo a cifrar y seleccionar la opción “AES Crypt”.
2. Introducir la clave en el cuadro de dialogo.
3. El archivo cifrado aparecerá con el mismo nombre que el original pero con la extensión *.aes*.

El archivo encriptado esta listo para, por ejemplo, enviarlo a un destinatario por correo u otro mecanismo, guardarlo en un pendrive, etc. Una consideración, si lo enviamos por correo, no enviar en el mismo mensaje el archivo y la clave. Debemos utilizar dos mensajes diferentes o canales para la transmisión del archivo y de la clave.

El proceso de descifrado es simple:

1) Pulsamos dos veces sobre el archivo .aes.
2) Introducimos la clave
3) El archivo desencriptado aparecerá con el nombre indicado pero sin la extensión .aes.

1https://www.aescrypt.com/

Ejercicio 1:
a) Instala la herramienta y cifra un archivo. Visualiza qué contiene el archivo cifrado.
b) Borra el archivo original. Ahora recupera el contenido original descifrando el archivo.

Nota: A partir de Windows 7, el propio sistema operativo de Windows suministra una herramienta denominada ***Bitlocker*** para realizar el cifrado, podemos acceder a ella desde el botón *Propiedades* de un archivo.

3.- Cifrado de dispositivos completos

Otra posibilidad de cifrado es mantener encriptada toda la información que se almacena en un dispositivo (disco, partición de disco, pendrive, tarjeta de memoria, CD, etc). De esta forma aunque alguien tenga acceso a dispositivo de almacenamiento, por ejemplo, perdida de un pendrive, le será imposible acceder a la información salvo que consiga la clave de cifrado.

Para ello, vamos a utilizar la herramienta **`VeraCrypt`**[2] que utiliza un algoritmo de cifrado AES-256[3]. En este caso, descargar y ejecuta el paquete **`VeraCrypt1.21.exe`**. Puede encontrar la documentación para su instalación y uso en la dirección web: https://securityinabox.org/es/guide/veracrypt/windows/.

Utilizaremos la herramienta para cifrar el sistema de archivos de un *pendrive*. En prácticas os daré un pendrive para realizarla, si lo haceis en casa antes de comenzar deberíais hacer una copia de seguridad de la información que tengáis en el pendrive en el disco pues el proceso borrará el contenido del mismo.

La ventaja de este método es que cualquier archivo que escribamos en el pendrive es automáticamente cifrado, o a la inversa, cada archivo que leamos de pen es descifrado de forma automática. Una vez desmontado el pendrive, su información es muy difícil de visualizar con lo que estará protegida frente al robo o pérdida.

Ejercicio 2:
a) Instalar la herramienta y cifrar un pendrive. Escribir algunos archivos en el.
b) Extraer el pendrive y volver a insertarlo ¿qué ocurre si no damos la clave?
c) Insertarlo, ahora dando la clave, y acceder a los archivos previamente guardados, ¿qué ocurre?

Una vez realizado el ejercicio, dejaremos el pendrive como estaba, es decir, lo devolvemos al formato original FAT[4] (File Allocation Table). Para ello solo debemos pincharlo en el sistema y cuando este lo detecte, lo seleccionamos y con el botón derecho de ratón y elegiremos la opción **Formatear**. Como formato elegimos FAT32 (Windows 10 usa NTFS[5] para los disco duros), y opcionalmente podemos poner una etiqueta al dispositivo (el nombre que aparecerá en el navegador de archivos cuando lo conectemos). Damos “Iniciar” y esperamos a que termine el proceso de

2 https://veracrypt.codeplex.com/.
3 https://es.wikipedia.org/wiki/Advanced_Encryption_Standard.
4 https://es.wikipedia.org/wiki/Tabla_de_asignaci%C3%B3n_de_archivos.
5 https://www.openstego.com/.

formateo que puede ser lento.

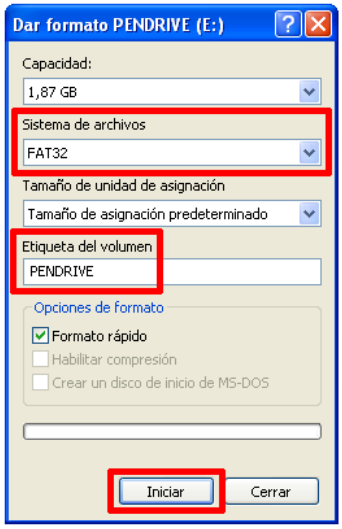

Figura 2.- Formateo de un dispositivo.

3.- Esteganografía

Un enfoque diferente de abordar el problema de hacer que cierta información no este accesible a personas no autorizadas es ocultar la información, no recodificarla como hace el cifrado. Esto es lo que hace la esteganografía: oculta un mensaje dentro de otro archivo que puede ser una imagen, video o audio. Para ver como funciona, vamos a utilizar la herramienta **openstego**[6]. Debemos ejecutar el archivo `Setup-OpenStego-0.7.1.exe`.

Para ocultar la información, contenida por ejemplo un archivo de texto, necesitamos un archivo portador. Para ello, podemos descargar de Internet una imagen (formatos .jpg, .bmp, .jpeg, .png, .ppm, o .tiff), de con buena resolución. Solo debemos ejecutar la herramienta con la opción “Hiden data” dándole como información: el archivo a ocultar, la imagen portadora, el archivo de destino y una clave. Esta herramienta nos producirá una nueva imagen que tiene oculto (los bits del archivo de texto se han mezclado con los píxeles de la imagen) el archivo que queríamos ocultar.

El proceso inverso se obtiene con la opción “Extract data” y consiste en extraer el contenido del archivo de texto de la imagen esteganográfica.

Ejercicio 3:
a) Crea un archivo de texto y busca una imagen de buena resolución. Utiliza la herramienta para ocultar el texto en la imagen.
b) Compara la imagen original con la nueva que ha generado la herramienta ¿hay diferencias visible?
c) Borra el archivo de texto original y utiliza ahora la herramienta para volver a recuperarlo.

3.- Hash de archivos

6 http://steghide.sourceforge.net/index.php.

Como indicábamos en el Apartado 1, las funciones *hash* sirven para garantizar la integridad de un archivo asegurando que el archivo no ha sido alterado desde que fue creado (desde su fuente original). Tiene diferentes usos como asegurar que un archivo descargo de Internet de no se ha alterando en el proceso o en análisis forense nos sirve para garantizar que una evidencia no ha sido modificada. También los podemos encontrar en las firmas digitales o en el almacenamiento de contraseñas.

En este apartado vamos a utilizar la herramienta *File Checksum Utility*[7] para calcular el hash de un archivo. La instalación de la misma no presenta dificultades. Para su uso solo tenemos que decidir si vamos a generar el hash de un único archivo o un directorio. Una vez decidido, le damos el nombre del archivo y el algoritmo a utilizar (elegiremos uno de los que nos ofrece la herramienta), tras lo cual la herramienta genera el hash.

> Nota: Actualmente, los algoritmos md5 y SHA-1 no se consideran seguros individualmente (se ha demostrado que se puede cambiar el contenido de un archivo y obtener el mismo hash para las dos versiones), pero algunas herramientas siguen usándolos combinados.

Ejercicio 4:
a) Generar los hashes, md5, sha-1 y sha-256, de un archivo de texto de vuestra elección y anotarlos.
b) Edita el archivo y cambia un único carácter del mismo. Vuelve a generar los hashes para el archivo modificado. Compara los hashes del archivo original y el modificado, ¿han cambiado?

7 http://www.blq-software.com/FileChecksumUtility/index-EN.html.