Grado en Ingeniería Informática

Departamento de Ingeniería Informática

Seguridad en los Sistemas Informáticos

Práctica 8: Criptografía Aplicada

Autores:
Juan Boubeta Puig,
Jesús Rosa Bilbao y
Jesús Lagares Galán

Fecha:
14 de diciembre de 2024

Índice

Índice de figuras

1. Objetivo

El objetivo de esta práctica es obtener un conocimiento general sobre criptografía aplicada en el ámbito de la seguridad informática en los tiempos actuales. Para ello, se darán a conocer algunos de los principales algoritmos de criptografía simétrica (usan una misma clave privada para cifrar y descifrar la información) y de criptografía asimétrica (usan un par de claves pública y privada).

Entonces se aprenderá a usar **OpenSSL** [1], una biblioteca de código abierto que proporciona implementaciones de protocolos criptográficos seguros. Además de ser una biblioteca, OpenSSL incluye una herramienta de línea de comandos que permite realizar varias operaciones criptográficas directamente desde la terminal como el cifrado y descifrado de datos, la creación de claves y la generación de *hash*, entre otros.

Asimismo, se conocerá qué es un *hash* y una firma digital y para qué sirven. Finalmente, se aprenderá a usar **GPG** [2], una herramienta de *software* libre muy útil para cifrar y descifrar, así como crear firmas digitales de forma simétrica y asimétrica.

2. Conocimientos previos

A continuación, se describen los conocimientos previos que han de conocerse para llevar a cabo esta práctica:

Criptografía Es el conjunto de técnicas empleadas para el cifrado de datos.

Criptoanálisis Es el conjunto de técnicas empleadas para la ruptura de los códigos criptográficos.

Criptología Se emplea para agrupar tanto a la Criptografía como al Criptoanálisis.

Esteganografía Ocultación en el interior de una información, aparentemente inocua, otro tipo de información.

Protocolo criptográfico Secuencia de mensajes que pueden estar cifrados, y que persiguen que la comunicación entre dos usuarios sea segura.

3. Cifrado simétrico

Es un tipo de cifrado que usa una misma clave para cifrar y descifrar mensajes en el emisor y el receptor. Las dos partes que se comunican han de ponerse de acuerdo de antemano sobre la clave a usar. Una vez que ambas partes tienen acceso a esta

clave, el remitente cifra un mensaje usando la clave, lo envía al destinatario, y este lo descifra con la misma clave.

3.1. DES

A continuación, se describe el algoritmo *Data Encryption Standard* (DES) y se presentan los ejercicios a realizar con este algoritmo.

3.1.1. Definición

DES es un algoritmo de cifrado simétrico (una misma clave debe ser conocida para cifrar/descifrar). Se utiliza como un método para cifrar información, cuyo uso se ha propagado ampliamente por todo el mundo. El algoritmo fue controvertido al principio, con algunos elementos de diseño clasificados, una longitud de clave relativamente corta, y las continuas sospechas sobre la existencia de alguna puerta trasera. Posteriormente DES fue sometido a un intenso análisis académico y motivó el concepto moderno del cifrado por bloques y su criptoanálisis.

Algunos comandos necesarios para la realización de esta parte de la práctica con la herramienta OpenSSL son los siguientes:

- Cifrar utilizando DES:
  `openssl enc -des -in plaintext.txt -out ciphertext.des`
- Descifrar utilizando DES:
  `openssl enc -des -d -in ciphertext.des -out decrypted.txt`

NOTA: Si al realizar estos comandos obtienes un error similar a "error setting cipher des cbc" prueba añadiendo a los comandos de cifrado y descifrado lo siguiente "-provider legacy -provider default".

3.1.2. Ejercicio 1

Cifre un texto con el algoritmo DES y envíelo por correo electrónico a un compañero. Los pasos a seguir son los siguientes:

1. Cifre el texto mediante el algoritmo DES usando la siguiente clave:
   *QaWsEdRfTgYh*13579*
2. Envíe el texto cifrado por email a un compañero.
3. Copie el texto cifrado que hemos recibido del otro compañero.
4. Descifre el texto cifrado con DES.

Indique el texto que ha elegido, su versión cifrada en DES, el compañero con el que ha intercambiado el texto cifrado, el texto que ha recibido en DES y el texto descifrado.

3.1.3. Ejercicio 2

Ahora que ya conocemos el algoritmo de cifrado DES, vamos a realizar algo más complicado:

- Bájese de internet una foto, que no sea NSFW, de un tamaño no muy grande.
- Cifre los datos mediante DES con la clave que desee.
- Envíe los datos cifrados por correo electrónico a un compañero (deberá comunicar a su compañero la clave que ha utilizado para realizar el cifrado).
- Descifre los datos que ha recibido de su compañero.

Indique la imagen que ha elegido, su versión cifrada en DES, el compañero con el que ha intercambiado la imagen, la imagen que ha recibido en DES y la imagen descifrada.

3.2. AES

A continuación, se describe el algoritmo *Advanced Encryption Standard* (AES) y se presentan los ejercicios a realizar con este algoritmo.

3.2.1. Definición

Al igual que DES, AES también es un algoritmo de clave simétrica (una misma clave debe ser conocida para cifrar/descifrar y tanto receptor como emisor deben tener una copia).

En este cifrado los datos a transmitir se dividen en bloques de 128 bits organizados en una matriz de cuatro por cuatro con cada byte en una posición de la misma. A continuación se toma la clave inicial y se utiliza para generar una serie de claves que se utilizarán en el proceso de cifrado (estas claves derivan de un método denominado **claves de Rijndal**). Una vez halladas las claves comienzan las rondas de encriptación:

1. Añadir llave redonda.
2. Bytes sustitutos.
3. Desplazar filas.

4. Mezclar columnas.
5. Agregar llave redonda (de nuevo).
6. Repetir los pasos anteriores unas 9, 11 o 13 veces. Esto dependerá si la clave es de 128, 192 o 256 bits.

De esta manera, un mensaje inicial podría ser:

```
Cómprame algunas papas fritas por favor
```

Se convierte en lo que parecería una cadena de caracteres completamente aleatoria, pero que tan solo ha sido cifrada utilizando AES:

```
Ok23b8a0i3j 293uivnfqf98vs87a
```

Algunos comandos necesarios para la realización de esta parte de la práctica con la herramienta OpenSSL son:

- Cifrar utilizando AES:
  `openssl enc -aes-256-cbc -in plaintext.txt -out ciphertext.aes`
- Descifrar utilizando AES:
  `openssl enc -aes-256-cbc -d -in ciphertext.aes -out decrypted.txt`

3.2.2. Ejercicio 3

Descargue del campus virtual el fichero "ciphertext.aes" y descifre el mensaje cifrado gracias al algoritmo AES sabiendo que la clave es: `qSZ*PPG6n3+ct+-'U5`

4. Cifrado asimétrico

En este tipo de cifrado se usa un par de claves para el cifrado/descifrado de mensajes. Una clave es pública y se puede compartir con cualquier persona, la otra clave es privada y el propietario debe guardarla de modo que nadie tenga acceso a ella. Además, los algoritmos criptográficos asimétricos garantizan que esa pareja de claves solo se pueda generar una vez, de modo que se asuma que no es posible que dos personas hayan obtenido casualmente la misma pareja de claves.

Para poder desarrollar correctamente los siguientes ejercicios, lo primero que deberá hacer es generar su par de claves pública y privada haciendo uso de la herramienta OpenSSL:

- Generar clave privada:
  `openssl genpkey -algorithm RSA -out private_key.pem`
- Generar clave pública correspondiente a la clave privada:
  `openssl rsa -pubout -in private_key.pem -out public_key.pem`

Una vez generadas las claves, deberemos almacenarlas en un lugar seguro, ya que son las que vamos a usar durante el desarrollo de los ejercicios. Conviene destacar que si las perdemos y volvemos a generar las claves estas serán distintas y, por lo tanto, lo que hayamos hecho anteriormente no valdrá.

4.1. RSA

A continuación, se describe el algoritmo Rivest, Shamir & Adelman (RSA) y se presentan los ejercicios a realizar con este algoritmo.

4.1.1. Definición

RSA es un sistema criptográfico de clave pública desarrollado en 1979. Es el algoritmo de este tipo más utilizado y es válido tanto para cifrar como para firmar digitalmente.

La seguridad de este algoritmo radica en el problema de la factorización de números enteros. Los mensajes enviados se representan mediante números, y el funcionamiento se basa en el producto conocido de dos números primos grandes elegidos al azar y mantenidos en secreto.

Algunos comandos necesarios para la realización de esta parte de la práctica haciendo uso de la herramienta OpenSSL son los siguientes:

- Cifrar utilizando la clave pública con RSA:
  `openssl pkeyutl -encrypt -pubin -in input_file.txt -out encrypted_file.enc -inkey public_key.pem -pkeyopt rsa_padding_mode:oaep`
- Cifrar utilizando la clave privada con RSA:
  `openssl pkeyutl -encrypt -in input_file.txt -out encrypted_file.enc -inkey private_key.pem -pkeyopt rsa_padding_mode:oaep`
- Descifrar utilizando la clave pública con RSA:
  `openssl pkeyutl -decrypt -pubin -in encrypted_file.enc -out decrypted_file.txt -inkey public_key.pem -pkeyopt rsa_padding_mode:oaep`
- Descifrar utilizando la clave privada con RSA:
  `openssl pkeyutl -decrypt -in encrypted_file.enc -out decrypted_file.txt -inkey private_key.pem -pkeyopt rsa_padding_mode:oaep`

- Firmar utilizando la clave privada con RSA:
  `openssl rsautl -sign -inkey private_key.pem -in input_file.txt -out signed_file.bin`
- Verificar firma utilizando la clave pública con RSA:
  `openssl rsautl -verify -pubin -inkey public_key.pem -in signed_file.bin -out verified_file.txt`

4.1.2. Ejercicio 4

Cifre un texto con el algoritmo RSA y envíelo por correo electrónico a un compañero. Los pasos a seguir son los siguientes:

- Solicite la clave pública a un compañero.
- Cifre el fichero de texto mediante el algoritmo RSA usando la clave pública de su compañero (la del receptor).
- Envíe el fichero cifrado en RSA por email a su compañero.
- Su compañero deberá descifrar el fichero con su clave privada (la del receptor).

Indique el texto que ha elegido, su versión cifrada en RSA, el compañero con el que ha intercambiado el texto, el texto que ha recibido en RSA y el texto descifrado. Además, conteste a la siguiente pregunta: ¿el cifrado que se ha llevado a cabo permite la confidencialidad o la autenticidad del mensaje enviado?

4.2. Ejercicio 5

Firme un texto con el algoritmo RSA y envíelo por correo electrónico a un compañero. Los pasos a seguir son los siguientes:

- Firme el fichero de texto mediante el algoritmo RSA usando su clave privada (la del emisor).
- Envíe el fichero firmado en RSA por email a un compañero.
- Su compañero deberá solicitarle la clave pública (la del emisor).
- Su compañero deberá verificar el fichero con la clave pública del emisor.

Indique el texto que ha elegido, su versión firmada en RSA, el compañero con el que ha intercambiado el texto, el texto que ha recibido en RSA y el texto verificado. Además, conteste a la siguiente pregunta: ¿el provceso que se ha llevado a cabo asegura la confidencialidad o la autenticidad del mensaje enviado?

4.2.1. Ejercicio 6

Ahora que ya conocemos el algoritmo de cifrado RSA, hagamos algo más complicado:

- Bájese de internet una foto, que no sea NSFW, de un tamaño no muy grande.
- Solicite la clave pública a un compañero.
- Cifre la imagen mediante el algoritmo RSA usando la clave pública de su compañero.
- Envíe la imagen cifrada con RSA por email a su compañero.
- Copie la imagen cifrada que ha recibido del compañero.
- Descifre la imagen con la clave privada suya.

NOTA: Como podemos comprobar, no es posible cifrar la imagen únicamente con la clave pública. Será necesario cifrar la clave para que tenga el tamaño adecuado. Las siguientes instrucciones serán de utilidad: `https://www.czeskis.com/random/openssl-encrypt-file.html`

Indique la imagen que ha elegido, su versión cifrada en RSA, el compañero con el que ha intercambiado la imagen, la imagen que ha recibido en RSA y la imagen descifrada.

5. Algoritmos *hash*

Seguidamente, se describe los algoritmos *hash* y se proponen varios ejercicios.

5.1. Definición

Una función criptográfica *hash* es un algoritmo matemático que transforma una entrada en una salida codificada con una longitud fija. A esta salida se le denomina en inglés *hash*, y huella o resumen digital en español. Esta función es unidireccional; es decir, utilizando el mismo algoritmo no se podría decodificar el mensaje. Independientemente de la longitud de los datos de entrada, el valor de salida tendrá siempre la misma longitud, podemos ver un ejemplo gráfico en la Figura 1. De este modo, una entrada como `Brian`, utilizando un algoritmo de *hashing*, se transformaría en `75c450c3f963befb912ee79f0b63e563652780f0`. Un pequeño cambio en el texto original genera una salida completamente diferente. Por ejemplo, utilizando el algoritmo MD5:

```
MD5("Generando un MD5 de un texto") = 5df9f63916ebf8528697b629022993e8
```

`MD5("Generando un MDS de un texto") = e14a3ff5b5e67ede599cac94358e1028`

Dentro de los diferentes algoritmos para crear *hashs*, destacamos los tres siguientes:

- MD5: Se utiliza usualmente en el traspaso de archivos para comprobar que algún archivo no haya sido modificado. Codifica con 128 bits y la salida se representa con 32 símbolos hexadecimales.
- SHA: Familia de funciones criptográficas, distinguimos entre SHA-0, SHA-1, SHA-2 y SHA-3. Cada tipo de SHA utiliza un número de bits para la codificación. Por ejemplo, SHA-0 utilizaba 160, SHA-1 utiliza 256, SHA-2 512 y SHA-3 1600.
- BlAKE: Presenta diferentes variantes en función del tamaño de palabra que utilice (32 bits y 64 bits). Utiliza una tabla de 16 palabras constantes y otra de permutaciones para realizar la codificación.

Este tipo de funciones son las que están detrás de muchos elementos que utilizamos en nuestro día a día como la firma digital. A su vez, gracias a este tipo de funciones criptográficas han podido nacer tecnologías emergentes como la *Blockchain*.

Algunos comandos necesarios para la realización de esta parte de la práctica con OpenSSL son:

- Calcular *hash* MD5 de un archivo:
  `openssl md5 archivo`
- Calcular *hash* SHA-256 de un archivo:
  `openssl sha256 archivo`

5.2. Ejercicio 7

Siga los siguientes pasos:

- Cree un archivo de texto llamado *documento.txt* con algún contenido.
- Utilice *openssl* para calcular el hash MD5 del archivo *documento.txt*.
- Guarde el *hash* MD5 resultante en un nuevo archivo denominado *documento_md5.txt*.
- Envíe a su compañero por email los archivos *documento.txt* y *documento_md5.txt* y pídale que, por un lado, calcule el *hash* MD5 del fichero *documento.txt* recibido y, por otro lado, compruebe si el *hash* obtenido es el mismo que el que ha recibido por correo.

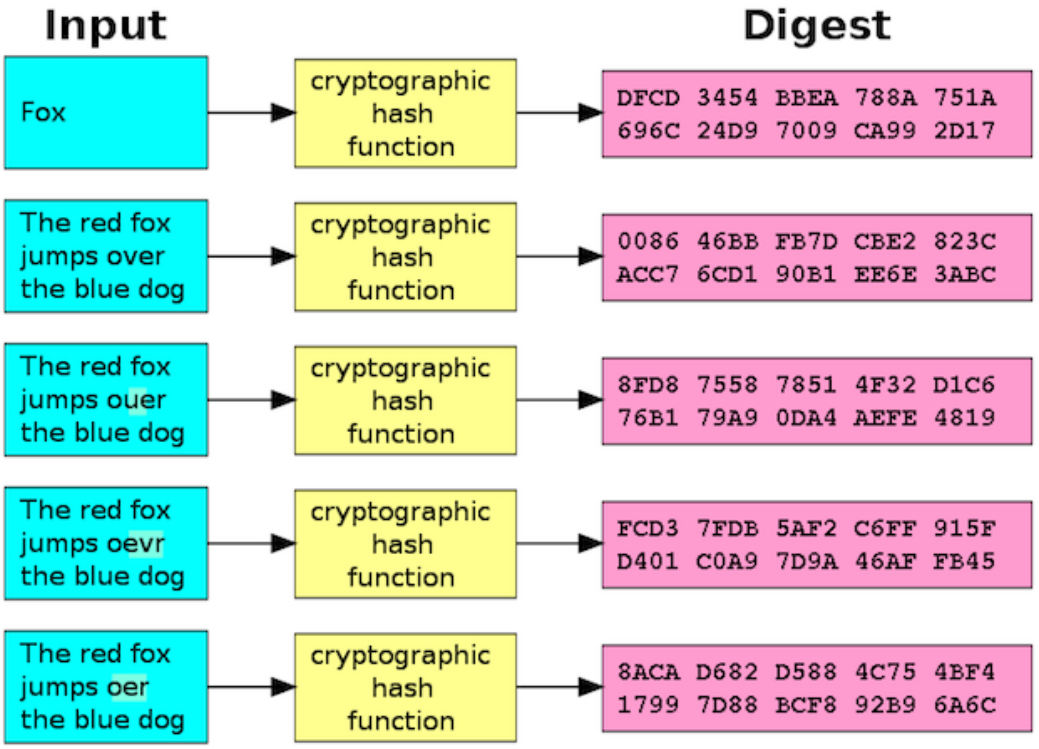

Figura 1: Función *hash* (extraída de Wikimedia Commons)

5.3. Ejercicio 8

Siga los siguientes pasos:

- Partiendo del mismo archivo de texto *documento.txt* creado en el ejercicio anterior, utilice *openssl* para calcular el hash SHA-256 de este archivo.
- Guarde el *hash* SHA-256 resultante en un nuevo archivo denominado *documento_sha256.txt*.
- Envíe a su compañero por email los archivos *documento.txt* y *documento_sha256.txt* y pídale que, por un lado, calcule el *hash* SHA-256 del fichero *documento.txt* recibido y, por otro lado, compruebe si el *hash* obtenido es el mismo que el que ha recibido por correo.

5.4. Ejercicio 9

Siga los siguientes pasos:

- Compare los dos archivos *documento_md5.txt* y *documento_sha256.txt*. ¿Son

diferentes? ¿Por qué sí o por qué no?

- Añada un espacio en blanco en cualquier lugar del contenido del fichero *documento.txt*.
- Calcule nuevamente los *hashes* MD5 y SHA-256 del *documento.txt*.
- ¿Son diferentes los *hashes* resultantes? ¿Por qué sí o por qué no?

6. Firmas digitales

A menudo escuchamos en nuestro día a día conceptos nuevos relacionados con la identidad como el certificado digital, la firma electrónica o la firma digital. No todos los conceptos son lo mismo, ni sirven para lo mismo, por eso es importante conocer y entender cada uno de ellos. En este caso veremos qué es una firma digital, para qué se utiliza y qué relación tiene con la criptografía actual.

6.1. Definición

Una firma digital (no un certificado digital) es una técnica criptográfica utilizada para validar la autenticidad e integridad de un mensaje, *software* o documento digital.

Se le asigna el nombre de "firma" porque es como conocemos en el mundo no virtual una forma de verificar la validez y veracidad de un documento (entre otras cosas). Por ejemplo, cuando obtenemos un certificado de alguna formación, este estará firmado por la institución correspondiente. Sin esta firma, el certificado tan solo sería un trozo de papel con un título; pero al estar firmado se convierte en una prueba (en la que la institución firma dando veracidad) de que hemos completado esa formación.

A diferencia de la firma tradicional del mundo físico, la firma digital no es una firma como tal, sino que es un nombre de un mecanismo que consta de dos "claves". Consiste en aplicar mecanismos criptográficos al contenido de un mensaje o documento para demostrar su autenticidad y, en algunos casos, su integridad.

Lograr una firma digital es el resultado de aplicar un ***hash*** (algoritmo matemático que devuelve siempre el mismo *output* siempre que se introduzca el mismo *input*, pero solo funciona en una dirección) a su contenido (mensaje, documento, etc.) y aplicar seguidamente el **algoritmo de firma** al resultado de dicho *hash*, generando de esta manera la firma electrónica.

6.2. Funcionamiento de la firma digital

Las firmas digitales se basan en la criptografía de clave pública o criptografía asimétrica. Normalmente, siempre que se da el proceso de una firma digital hay 3 algoritmos criptográficos involucrados:

- **Generación de claves**. Un algoritmo proporciona una clave pública junto a su clave privada.
- **Firma**. Un segundo algoritmo produce una firma al recibir una clave privada junto a un mensaje.
- **Verificación**. El último algoritmo comprueba la autenticidad del mensaje al verificarlo con la firma y la clave pública.

Y el proceso en el que trabajan todos estos algoritmos es el siguiente:

1. Se crea un **hash** de la información que se le ha dado (mensaje, documento, etc.).
2. Se cifra el **hash** utilizando la clave privada, por lo que se va a transmitir un **hash** cifrado. Junto a este **hash** cifrado se transmite otra información que es la firma digital. Gracias a este mecanismo conseguimos la integridad de la información, ya que cualquier cambio en los datos daría como resultado un valor de *hash* diferente.
3. Cuando la información llega hasta el receptor este utiliza la clave pública para descifrar el **hash**. Si los dos **hashes** coinciden, tenemos la prueba de que los datos no han cambiado desde que se firmó; además, aseguramos también que la clave privada con la que se ha "firmado" no se ha visto alterada y es la que corresponde con la clave pública (véase la Figura 2).

De esta manera, las firmas digitales dificultan que el firmante niegue no haber firmado algo, ya que la única opción sería que su clave privada se hubiese visto comprometida, y verifica la integridad de la información traspasada.

Para utilizar estas firmas en nuestro equipo contamos con diversas herramientas como GnuPG (GPG).

6.3. GPG

GPG es una herramienta de *software* libre empleada para comunicaciones seguras y almacenamiento de datos. Se puede usar tanto para cifrar como para crear firmas digitales de forma simétrica y asimétrica.

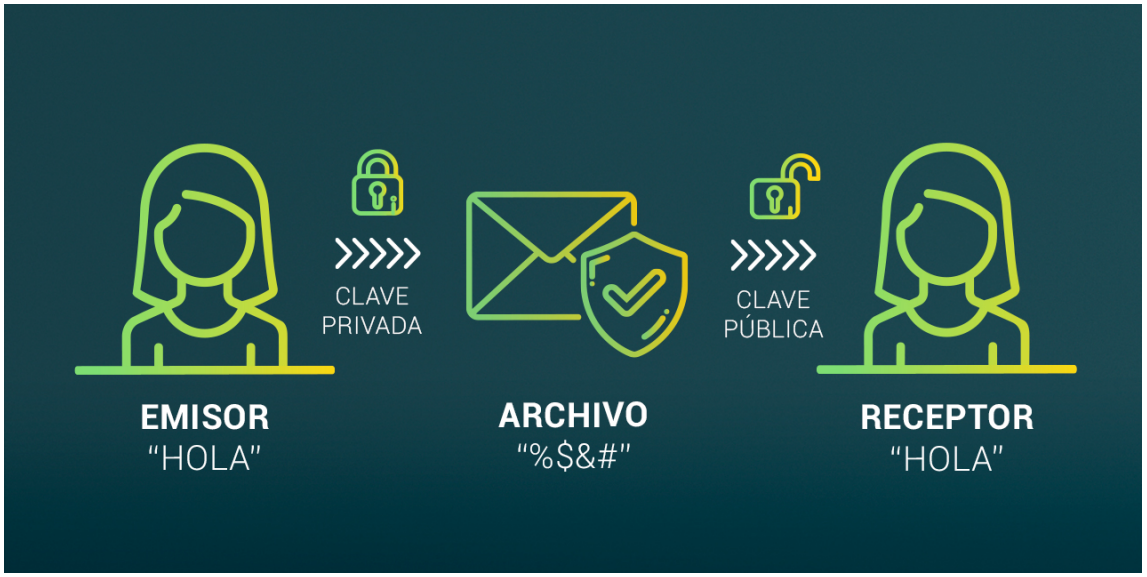

Figura 2: Representación del proceso de firma digital, extraída de [3]

Para obtener autenticidad, crea una huella o resumen digital del mensaje (en inglés, *hash*), y lo cifra con la clave privada, usando un algoritmo de cifrado asimétrico (normalmente **ElGamal**, aunque también puede usar **DSA** o **RSA**). Posteriormente, el destinatario puede comprobar que el resumen calculado a partir del mensaje recibido coincide con el descifrado.

Para obtener confidencialidad, utiliza un esquema híbrido, en el que un algoritmo asimétrico (**ElGamal** por omisión) cifra una clave para un algoritmo simétrico (**AES**, entre otros), que es el que realmente cifra el fichero indicado.

También podemos generar nuestro par de claves pública y privada gracias al siguiente comando proporcionado por la herramienta GPG:
`gpg --gen-key`

Al ejecutar, por primera vez, este comando se creará el directorio *.gnupg* con el fichero de configuración y los ficheros donde se almacenarán las claves privadas y las claves públicas. Dentro de nuestra máquina Kali Linux encontraremos este fichero en `root/.gnupg`

Para utilizar la herramienta tan solo tendremos que hacer uso del comando `gpg` junto a alguna de sus opciones. Por ejemplo:

- Visualizar la lista de claves privadas (cifrado asimétrico) disponibles:
  `gpg --list-secret-keys`

- Visualizar la lista de claves públicas (cifrado asimétrico) disponibles:
  `gpg --list-keys`
- Cifrar de forma simétrica, por ejemplo un archivo `texto.txt`:
  `gpg -c texto.txt`
- Descifrar de forma simétrica, por ejemplo un archivo `texto.txt`:
  `gpg -d texto.txt.gpg`

Cabe destacar que, si fuese necesario borrar alguna clave, en primer lugar habría que borrar la clave privada y, seguidamente, la pública:

```
gpg --delete-secret-key identificador_clave
gpg --delete-key identificador_clave
```

6.3.1. Firma y cifrado

A continuación, se dan las instrucciones generales para cada acción:

- Firmar un texto en claro con su clave privada, dejando los contenidos visibles y codificando la firma en Base-64 («ASCII armored»):

  `gpg -o fichero_firmado --clearsign fichero`

- Verificar la firma incrustada en un fichero:

  `gpg --verify fichero_firmado`

- Cifrar un fichero para una serie de destinatarios que se pedirán en la línea de órdenes, codificándolo en Base-64:

  `gpg -a -o fichero_cifrado --encrypt fichero`

- Cifrar y firmar un fichero, codificando el resultado en Base-64:

  `gpg -o fichero_cifrado_y_firmado -a --sign --encrypt fichero`

- Descifrar el fichero, comprobando su firma, si la tiene:

  `gpg --decrypt fichero_cifrado_y_opcionalmente_firmado`

6.3.2. Ejercicio 10

Responda a la siguiente pregunta: ¿por qué cuando hablamos de certificado digital no estamos hablando expresamente de firma digital?

6.3.3. Ejercicio 11

Utilizando la herramienta GPG (véase Sección 6.3):

1. Sabiendo que la contraseña con la que se ha cifrado el archivo proporcionado en la práctica (*ejercicio11.txt.gpg*) es `ZANXAz7:fr1r4t4*>8`, descífrelo y responda: ¿se ha cifrado de forma simétrica o asimétrica? ¿Qué comando ha utilizado para ello?

2. Cree un archivo `.txt` en el que escriba qué es lo que aparece escrito en el archivo `ejemplo.txt` descifrado en el apartado anterior y cífrelo de forma simétrica utilizando para ello la contraseña `simetría`. ¿Qué comando ha utilizado para cifrar? ¿Lo ha hecho de forma simétrica o asimétrica? ¿Por qué?

6.3.4. Ejercicio 12

Nuevamente, utilizando la herramienta GPG:

1. A partir de las claves públicas y privadas generadas, cifre un fichero PDF cualquiera con su clave privada.

2. Envíele a un compañero suyo tanto el fichero cifrado como su clave pública.

3. El compañero deberá descifrar el fichero cifrado, a partir de la clave pública compartida (esta clave pública deberá importarse previamente en GPG).

4. Comentar y mostrar todo el proceso seguido.

Referencias

[1] OpenSSL: *OpenSSL*. `https://www.openssl.org`, visitado el 04/12/2023.

[2] The GnuPG Project: *GPG*. `https://www.gnupg.org/`, visitado el 04/12/2023.

[3] Leyre Soto: *¿Qué es una firma digital?* `https://blog.signaturit.com/es/que-es-una-firma-digital#3`, visitado el 04/12/2023.