Conceptos sobre Seguridad

Hash: Algoritmos que através de una entrada(ya sea texto, contraseña o un archivo) devuelven una cadena alfanumérica que una longitud normalmente fija representando una especie de "resumen" de la cadena entrante.(Esa cadena solo se repetirá con la misma entrada de datos).

Cifrado: 2 tipos:

```
1.asimétrico: 1 clave pública para descifrar y otra privada para cifrar.
2.simétrico : 1 sola clave para encriptar y descifrar.
```

Firma electrónica: En sencillos pasos:

```
1º. Aplicación de Hash sobre el documento inicial.
(Ejemplo: fichero.txt generaría fichero.hash)

2º. Cifrado con clave privada al "fichero.hash".

En resumen, siendo nosotros el cliente recibimos el fichero original y
el fichero hash correspondiente encriptado previamente con la clave privada;

Para la autenticación del fichero aplicaríamos el hash sobre el original creando
un nuevo hash(Ejemplo: ficheroComprobacion.hash), desencriptamos el fichero.hash,
recibido junto al fichero original, mediante la clave pública y comparando
ambos ficheros(fichero.hash y ficheroComprobacion.hash) podremos saber si
el archivo es completamente original.
```

OpenSSL

Cifrado con HASH:

Teniendo 2 ficheros como objetivos de prueba: prueba1 y prueba2

Contenido fichero1:

```
Comprobacion hash1
```

Contenido fichero2:

```
Comprobacion hash2
```

1. Realizamos sobre prueba1 y prueba2:

   openssl dgst -sha -out prueba1.hash prueba1.txt

2. Podemos comprobar que el resultado de la encriptación es completamente diferente:

   2.1 Resultado prueba1.hash:

   ```
   SHA(prueba1.txt)= 918c91430796b7792c7cf5165022da9442e404f7
   ```

   2.2 Resultado prueba2.hash:

   ```
   SHA(prueba2.txt)= f96cea198ad1dd5617ac084a3d92c6107708c0ef
   ```

3. Podemos comprobar como cambiando un único carácter el código generado es completamente diferente.

Cifrado con algoritmo simétrico:

En este caso, veremos un ejemplo de como cifrar un archivo utilizando una encriptación simétrica, usamos:

1. Cifrado simétrico:

   ```
   openssl enc -aes-128-cbc -in archivoParaEncriptar -out archivoEncriptado.enc

   Contenido del archivo: Prueba encriptacion simetrica.

   *NOTA: Justo después nos pedirá la contraseña que habrá que introducir después
   para la correcta desencriptación.

   Resultado: Salted__=��h@V�r�j&&w~4
   ```

2. Descifrado simétrico:

   ```
   openssl enc -d -aes-128-cbc -in archivoEncriptado.enc -out archivoDesencriptado.c

   Como salida obtendríamos el contenido de archivoParaEncriptar en archivoDesencrip
   ```

Cifrado con algoritmo asimétrico

- Generación de clave privada:

```
openssl genpkey -algorithm RSA -out privatekey

·Ejemplo de resultado de privatekey:
-----BEGIN PRIVATE KEY-----
MIICdwIBADANBgkqhkiG9w0BAQEFAASCAmEwggJdAgEAAoGBAM9M9CbjQ5spSr19
7yqx5pUp/K/hlZrlvec2ZOq8uwslaNFobnKE75jpW+Ow7FWqPGyQn/W3tEJ5RSqF
GIyPD0J6pO+zY3ZAFs6CinAEtXpSgfVHMBAkW0DwKnhmhkYc7Pdy9RUs27SSETXR
nxVKEz8p2HHz/QtmuT7/aLyoNp9NAgMBAAECgYEAxBcuCK8Hn3lCytaaaw+XZ1gt
P44q/zf9F/5bHbTjVZrQeO3JNCjNoB52Hn6YQDxlJej0dvGL4SlnpN37p9IXgnz8
DO22EzF8bdgu1XcywxAS5l35TxuP5RJ1Pdj68psZcXkaIxSc+QszbP3CGodwvfYN
WvvMKrnhOBkgqLFR+J0CQQD4l+aBubrJX3UHTxpDCF0KnWFiGUZgckR/trphHwcl
sA+IH+7vi3WjSu4efgCYkl2UlFBiI8Vf7rr514tafPvHAkEA1XoZt5QZJbjG6Aue
vb1ohm0QZ2abXNVsCqiflOd7IeMdqiQ5cVUg2T3LlCDxgPB92jcOlwSrFEyEPCwq
R4hESwJAB1rfLAlluF2FpeC2QzN1JUJAxlk/Fs1qr2ilnQgA+yF5ZCqltBqpMVqf
LvHFbL4v3JyCwFSUm2EB0TCKY/P8swJAKIV8zDP4cgiY+QL7ptiFNr8NFhOL+hso
3v0TDZjC7rPFtb4aiUC3c4UfV4bJUMhVm05IHrJsJvY/Q4y+mcVr8QJBAMvoyjS+
zgTxB3pl2znY/3v5gyPtsikh9bBq4Gz0wrsgoPzYi9Z2Sz1mxtaCcWlkRoWQ69Rr
VEPxvx+2EVuxAg4=
-----END PRIVATE KEY-----

·Descripción de parámetros: -"privatekey" es el archivo resultante, contenedor
                            -"RSA" es el algoritmo elegido para encriptar.
```

- Generación de la clave pública:

```
openssl pkey -in privatekey -pubout -out publickey

·Ejemplo de resultado de publickey:
-----BEGIN PUBLIC KEY-----
MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQDPTPQm40ObKUq9fe8qseaVKfyv
4ZWa5b3nNmTqvLsLJWjRaG5yhO+Y6VvjsOxVqjxskJ/1t7RCeUUqhRiMjw9CeqTv
s2N2QBbOgopwBLV6UoH1RzAQJFtA8Cp4ZoZGHOz3cvUVLNu0khE10Z8VShM/Kdhx
8/0LZrk+/2i8qDafTQIDAQAB
-----END PUBLIC KEY-----

·Descripción de parámetros: -"privatekey" archivo entrante, contenedor de la cl
                            a partir de ella se genera la clave pública.
                            -"publickey" archivo resultante, contenedor de la c
```

1. Cifrar con clave pública

```
openssl pkeyutl -pubin -encrypt -in fichero -out fichero.enc -inkey publickey

·Contenido de fichero:
  Archivo para cifrar con clave pública


·Resultado:
```

```
  p4���z@
  K/w@{D?@@5@twN@_@ @P@^@%q@@@@+ɹ@
  @r@U''@@y@®@k@\E@@@@)@

·Descripción de parámetros: -"fichero" el archivo a cifrar
                            -"fichero.enc" el fichero donde almacenar la encriptació
                            -"publickey" es el fichero que contiene nuestra clave pub
```

1.1.Descifrar con la clave privada encriptada previamente con pública

```
openssl pkeyutl -decrypt -in fichero.enc -inkey privatekey -out fichero.desc

·Contenido de fichero:
  p4���z@
  K/w@{D?@@5@twN@_@ @P@^@%q@@@@+ɹ@
  @r@U''@@y@®@k@\E@@@@)@

·Resultado:
  Archivo para cifrar con clave pública

·Descripción de parámetros: -"fichero.enc" contiene la encriptación generada con la c
                            -"privatekey" contiene nuestra clave privada que utilizam
                            -"fichero.desc" contiene el resultado del archivo cifrado
```

Firmar

```
openssl pkeyutl -sign -in prueba.txt -out prueba.sig -inkey privatekey

·Contenido de prueba.txt:
  Prueba de Firmar

·Salida:
  7pu@(@+    @V@Bh/@@@H@aP0@Q@@$M@c@@_:ŸSk8;@@0@@EjVC@?@@d@@@@@u@@@@]%`@@Ez:@@@@@‘u@H

·Descripción de parámetros: -"privatekey" es la clave privada
                            -"prueba.txt" el archivo al que aplicamos la firma
                            -"prueba.sig" resultado firmado
```

Verificación de la firma

```
openssl pkeyutl -pubin -verify -sigfile fichero.sig -in fichero(inicial) -inkey publi

·Contenido de prueba.sig
  7pu@(@+    @V@Bh/@@@H@aP0@Q@@$M@c@@_:ŸSk8;@@0@@EjVC@?@@d@@@@@u@@@@]%`@@Ez:@@@@@‘u@H

·Salida:
```

```
  Signature Verified Successfully

·Descripción de parámetros: -"publickey" nuestro contenedor de clave publica
                            -"fichero" es el fichero original a comparar
                            -"fichero.sig" es el fichero firmado que comparamos con e
```