Actividad 5: Cifrado simétrico con openSSL

Seguridad Informática

23/10/2014

Brais López Yáñez

ÍNDICE

TRABAJO POR PAREJAS

Esta tarea se ha realizado con el apoyo de Manuel Cascallar Autrán.

Ejercicio: Codificar un fichero de texto que contenga una contraseña arbitraria en base64, y enviadlo a vuestro compañero. El compañero debe obtener la contraseña a partir de este fichero.

1. **Lo primero que hice fue encriptar mi contraseña e enviársela a través del correo:**
   Openssl enc –base64 –in contrasinal.txt –out cifrada
2. **Lo siguiente, tras recibir la contraseña cifrada de Manuel, la desciframos con el siguiente comando:**
   Openssl enc –base64 –d –in cifr –out descifradaManu

Usad la contraseña como clave para cifrado/descifrado. Cada uno de vosotros debe cifrar un texto plano usando DES en modo cbc, enviar el fichero cifrado al otro, y descifrar el fichero que recibís.

3. **Cifrar fichero de texto, con la contraseña de mi compañero:**
   Openssl des-cbc –in Texto –out textoProtegido
4. **Desciframos el texto que nos manda Manuel, con mi contraseña:**
   Openssl des-cbc –d –in archivo –out archivoManu

INDIVIDUALMENTE

CIFRADO/DESCIFRADO

- Usad DES en modo cbc para cifrar un texto pequeño, y un texto más grande. Comprobad el valor del vector de inicialización en cada caso (añadir la opción -p al cifrar). ¿Hay alguna diferencia? ¿Para qué se utiliza este vector de inicialización IV?

Contraseña texto breve

Openssl des-cbc –p –in textoBreve –out textoBreveCif

Verifying - enter des-cbc encryption password:

salt=AB126263D98146ED

key=7E51BD104A974312

iv =360ED92F86A5616F

Contraseña texto grande

Openssl des-cbc –p –in textoGrande –out textoGrandeCif

enter des-cbc encryption password:

Verifying - enter des-cbc encryption password:

salt=86C3A481052D0B14

key=76E0EEB68484F63C

iv =E04E015AD92460A5

El password usado es el mismo para los dos casos, el contenido de las IV es distinto pero de igual tamaño.

password=1234

El IV es un bloque de bits que es requerido para permitir un cifrado en flujo o un cifrado por bloques, en uno de los modos de cifrado, con un resultado independiente de otros cifrados producidos por la misma clave. El tamaño del IV dependen del algoritmo de cifrado y del protocolo criptográfico y a menudo es tan largo como el tamaño de bloque o como el tamaño de la clave.

- Comparad los resultados del cifrado usando diferentes algoritmos simétricos (des, 3des, aes, ...), fijándoos en la longitud de la clave, el tamaño del bloque, el tamaño del fichero cifrado, los vectores de inicialización,...

AES-128

computer@Computer:~/Seguridad5$ openssl aes-128-cbc -p -in textoGrande -out textoGrandeCifAES

enter aes-128-cbc encryption password:

Verifying - enter aes-128-cbc encryption password:

salt=C92A4C5BD2EFB602

key=02FE9293BFDDE83BC8A4A99C23BAD042

iv =F1476AD9A2B1266ED0D689534B3685EB

DES

Openssl des-cbc –p –in textoBreve –out textoBreveCif

Verifying - enter des-cbc encryption password:

salt=AB126263D98146ED

key=7E51BD104A974312

iv =360ED92F86A5616F

DES3

computer@Computer:~/Seguridad5$ openssl des3 -p -in textoBreve -out textoBreveCifDES3

enter des-ede3-cbc encryption password:

Verifying - enter des-ede3-cbc encryption password:

salt=11C0CACE4D3A8BCA

key=69485B6F0599BA73A8BC597C3F16F68C2ED7EEAD8F14AC18

iv =228005E455462E2F

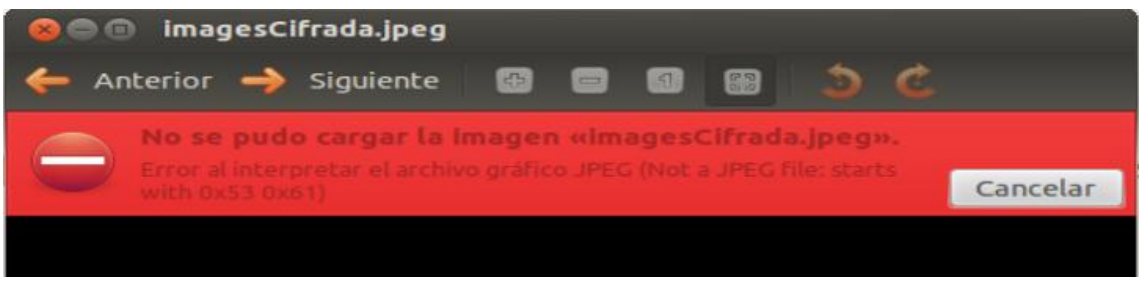
Una vez cifrados los archivos con diferentes tipos de cifrado podemos concluir que des3 es más seguro que des, ya que es 3 veces más grande. Por lo tanto utiliza más bits para cifrar. El algoritmo aes, es mejor que el des ya que es la continuación mejorada de este. Este algoritmo mejorado es más eficiente y más veloz que el anterior.

- Aplicad un algoritmo de cifrado simétrico a un fichero más complejo (una imagen, un fichero pdf, ...). Comprobad que el fichero cifrado no puede abrirse con la aplicación original.

Imagen cifrada:

Openssl des-cbc –p –in images.jpeg –out imagesCifrada.jpeg

Comprobación de la apertura de la imagen cifrada.



- Averiguad y explicad cómo se utiliza el rellenado ("padding") en los algoritmos de cifrado simétrico.

  El **Padding Scheme**, traducido como "Esquema de relleno", es un método de "camuflaje" que se añade a las funcionalidades de encriptación de RSA para ayudar a que la codificación no sea debil ni vulnerable. Básicamente, lo que hace el Padding Scheme es añadir bits de más en el mensaje m de forma que al ser cifrado se camufle el mensaje cifrado junto a los bits añadidos.

EXTRACTO DIGITAL (HASH)

- Obtened el resumen o extracto digital (hash) de un texto plano usando el algoritmo MD5.

  computer@Computer:~/Seguridad5$ openssl dgst -md5 textoBreve

  MD5(textoBreve)= 8be7b5c5515b4830d8d8cc88baed3cbc

- Modificad un único carácter en el texto y obtened el resumen de nuevo. Comprobad qué sucede con el extracto (longitud, contenido). ¿Para qué puede utilizarse un extracto digital?

  computer@Computer:~/Seguridad5$ openssl dgst -md5 textoBreve

  MD5(textoBreve)= ec286d53e474a77c30e20ad83e62af94

  computer@Computer:~/Seguridad5$

El contenido cambia, pero la longitud es la misma.

El extracto digital se usa para saber que un documento está integro tras su recepción, por eso se usa para comprobar que un archivo se a descargado correctamente o para comprobar que datos como un pequeño texto sigue siendo el mismo tras su emisión.

- Obtened el resumen de un texto plano más grande, y de una imagen, usando el algoritmo MD5. Comparad las longitudes de los extractos.

MD5

computer@Computer:~/Seguridad5$ openssl dgst -md5 textoGrande

MD5(textoGrande)= b7661ff9e7e54d7503805db2bffdad52

computer@Computer:~/Seguridad5$ openssl dgst -md5 images.jpeg

MD5(images.jpeg)= 143e470140e9938530a9b1db4610a58d

La longitud de los extractos es la misma, ambos tienen 32 caracteres. Que equivalen a 128 bits.

- Obtened el extracto digital de los ficheros empleados en los puntos anteriores, pero ahora utilizando el algoritmo SHA1. ¿Cuáles son las diferencias?

  - *computer@Computer:~/Seguridad5$ openssl dgst -sha1 images.jpeg*
    SHA1(images.jpeg)= 0c9f031d420fba211a749d55b990e82af842ec2d
  - *computer@Computer:~/Seguridad5$ openssl dgst -sha1 textoGrande*
    SHA1(textoGrande)=
    1c151b4f10ca092e64974bc3e749e2aea5f91222
  - *computer@Computer:~/Seguridad5$ openssl dgst -sha1 textoBreve*
    SHA1(textoBreve)= 4008a6e7d3479554df6f95b6fbf2a81837ecd7ed

La longitud de los extractos es mayor en el algoritmo SHA1, que equivalen a 160 bits y la de MD5, como comentamos antes, a 128 bits.

CIFRADO SIMÉTRICO EN HTTPS

- ¿Qué algoritmo de cifrado simétrico se usa para cifrar la conexión en Moodle? ¿Y en Gmail y en Correoweb? Comprobad otros sitios web públicos como Paypal, bancos, tiendas virtuales,... y si es posible, utilizando diferentes navegadores.

Moodle

- **Google Chrome:** La conexión utiliza TLS 1.2. La conexión se ha encriptado y autenticado con AES_128_GCM, y utiliza ECDHE_RSA como el mecanismo de intercambio de clave.
- **Firefox:** Conexión cifrada: cifrado de grado alto (AES-128, claves de 128 bits)

Gmail

- **Google Chrome:** La conexión utiliza TLS 1.2. La conexión se ha encriptado y autenticado con AES_128_GCM, y utiliza ECDHE_RSA como el mecanismo de intercambio de clave.
- **Firefox:** conexión cifrada: cifrado de grado alto (RC4, claves de 128 bits)

Correoweb

- **Google Chrome:** La conexión utiliza TLS 1.0. La conexión se ha encriptado y auntenticado con AES_128_CBC, con SHA1 para autentificación del mensaje y DHE_RSA como el mecanismo de intercambio de clave.
- **Firefox:** conexión cifrada: cifrado de grado alto (AES-256, claves de 256)

Paypal

- **Google Chrome:** La conexión utiliza TLS 1.2. La conexión se ha encriptado con RC4_128, con SHA1 autenticación mensaje y RSA como el mecanismo de intercambio de clave.
- **Firefox:** Conexión cifrada: cifrado de grado alto (RC4, claves de 128 bits)

Banco Popular

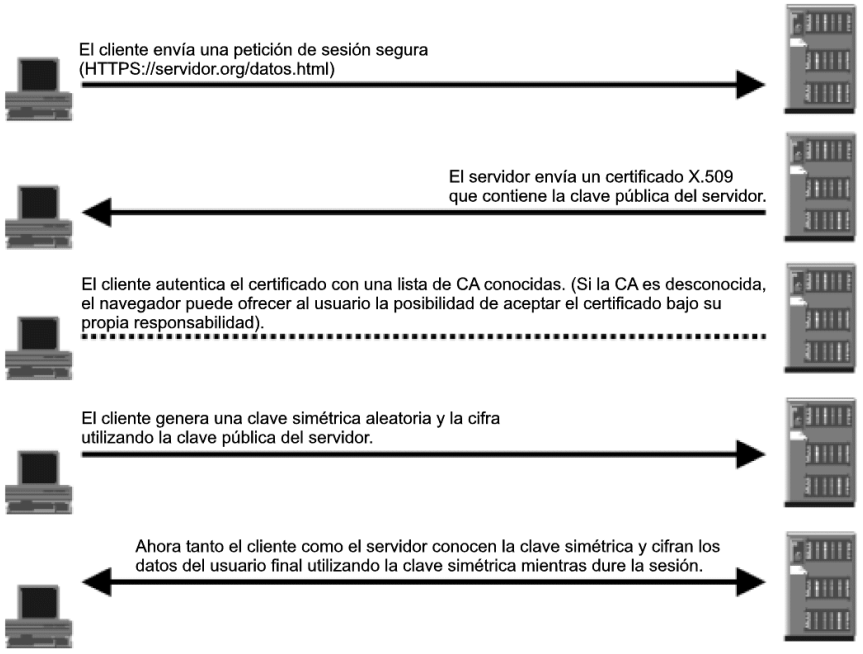
- **Google Chrome:** La conexión utiliza TSL 1.2. La conexión se ha encriptado con RC4_128, ha utilizado SHA1 para autenticación del mensaje y RSA como el mecanismo de intercambio de clave.
- **Firefox:** conexión cifrada: cifrado de grado alto (RC4, claves de 128 bits)

- Averiguad y explicad cómo se selecciona el algoritmo simétrico a utilizar en una conexión particular.

Algoritmo simétrico: El cifrado mediante clave simétrica significa que dos o más usuarios, tienen una única clave secreta, esta clave será la que cifrará y descifrará la información transmitida a través del canal inseguro.

A la hora de cifrar los datos ,que transcurren entre el servidor y el cliente, se utiliza un algoritmo simétrico como DES o RC4. Los algoritmos de clave pública, se utilizarían para el intercambio de las claves de cifrado y para las firmas digitales. Además de esto, el servidor utiliza un algoritmo de clave pública para el certificado digital. Con este certificado, el cliente, verifica la entidad del servidor.

Procedimiento:



FUENTES

- *http://www.redeszone.net/2010/11/04/criptografia-algoritmos-de-cifrado-de-clave-simetrica/*
- *https://publib.boulder.ibm.com/tividd/td/TRM/SC23-4822-00/es_ES/HTML/user277.htm*
- *http://es.wikipedia.org/wiki/Vector_de_inicializaci%C3%B3n*
- *http://www.genbetadev.com/seguridad-informatica/que-son-y-para-que-sirven-los-hash-funciones-de-resumen-y-firmas-digitales*