AES, OpenSSL y modo de operación ECB

Objetivos

- Familiarizarse con el uso de la biblioteca OpenSSL, particularmente con las funciones para cifrar archivos empleando un algoritmo de cifrado por bloques como AES.
- Identificar la importancia de seleccionar un adecuado modo de operación para el cifrado simétrico por bloques, en particular para con el algoritmo AES.

Requisitos

- Contar con la biblioteca OpenSSL instalada
- Se sugiere descargar y usar este archivo para los ejercicios de la Parte 1.
- Descargar de esta carpeta los archivos de la subcarpeta que le corresponda (ver Parte 2 de la práctica).

PARTE 1: AES con OpenSSL

Introducción

En los algoritmos de cifrado simétricos por bloques, el mensaje es cifrado agrupando o formando bloques de datos del mensaje original, de manera que se van cifrando uno a uno los bloques de datos de tamaño constante, de acuerdo con el algoritmo, y utilizando para ello una llave.

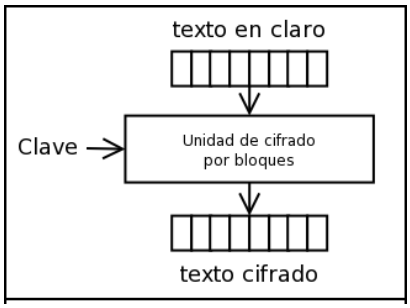

Fig. 1 Cifrado por bloques

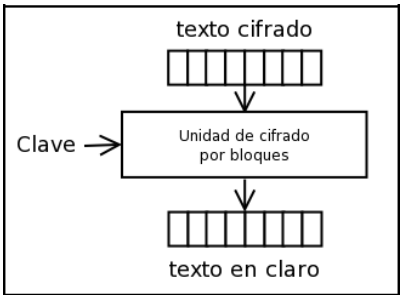

Fig. 2 Descifrado por bloques

Cifrado y modos de operación

Un modo de operación es una técnica que se usa para poder aplicar un algoritmo criptográfico a un conjunto de datos cuya longitud es mayor a la del tamaño de bloque del algoritmo en cuestión.

El estándar FIPS 81 establece 4 modos de operación[1]:

- ECB – Electronic Code Book
- CBC – Cipher Block Chain
- CFB - Cipher Feedback
- OFB – Output Feedback

Desarrollo

1. Cifrar un archivo con AES-128. Se recomienda usar el archivo `mensaje.txt`.
   Cifrar el archivo con el siguiente comando:

   ```
   $ openssl enc -v -aes-128-ecb -K 6e4b27a9a3a32165eb29eff45c204b0a -in \
     mensaje.txt -out mensajePadding.enc
   ```

   Con este comando se cifra el archivo *mensaje.txt*, con el algoritmo AES con una llave de 128 bits en el modo de operación **ECB** y se guarda el archivo cifrado en el archivo *mensajePadding.enc*. Observe que la llave se indica en el mismo comando.

   ```
   cripto@lab:~/aes$ openssl enc -v -aes-128-ecb -K 6e4b27a9a3a32165eb29eff45c204b0a -in mensaje.txt -out mensajePadding.enc
   bufsize=8192
   bytes read   :      118
   bytes written:      128
   ```

   Fig. 3 Cifrado de un archivo con OpenSSL

   La salida del comando indica que fueron leídos 118 bytes del archivo con el mensaje en claro, y que se escribieron 128 bytes en el archivo cifrado, es decir, 10 bytes extras. Esto se debe a que AES cifra por bloques de 128 bits o 16 bytes,

   $$\frac{118\ bytes}{16\ bytes}\frac{mensaje\ en\ claro}{tamaño\ de\ bloque\ en\ AES} = 7\ bloques\ de\ 16\ bytes, con\ residuo\ de\ 6\ bytes$$

   por lo que, para completar el último bloque de 6 bytes, OpenSSL añade 10 bytes de relleno o *padding*, de esta forma se tienen

   $$\frac{118 + 10\ bytes}{16\ bytes} = 8\ bloques\ de\ 16\ bytes$$

[1] Tenga en cuenta que estos no son los únicos modos de operación existentes.

2. Podemos comprobar su tamaño (en bytes) listando los archivos con el comando `ls -l`.

```
cripto@lab:~/aes$ ls -l
total 8
-rw-r--r-- 1 cripto cripto 128 oct  5 02:03 mensajePadding.enc
-rwxr-xr-x 1 cripto cripto 118 oct  5 01:58 mensaje.txt
```

Fig. 4 Verificación del tamaño de los archivos

3. Cifrar nuevamente el archivo *mensaje.txt*, pero ahora indicándole a OpenSSL que no agregue los bytes de relleno o *padding*, guardando el resultado en el archivo *mensajeNoPadding.enc*, esto se hace con el comando,

```
$ openssl enc -v -aes-128-ecb -K 6e4b27a9a3a32165eb29eff45c204b0a -in \
  mensaje.txt -nopad -out mensajeNoPadding.enc
```

OpenSSL marcará un error que indica que el tamaño del archivo no es múltiplo de 128 bits o 16 bytes, aun así, cifra los bloques que sí completan el tamaño requerido de 16 bytes.

```
cripto@lab:~/aes$ openssl enc -v -aes-128-ecb -K 6e4b27a9a3a32165eb29eff45c204b0a -in mensaje.txt -nopad -out mensajeNoPadding.enc
bufsize=8192
bad decrypt
140642520016000:error:0607F08A:digital envelope routines:EVP EncryptFinal ex:data not multiple of block length:../crypto/evp/evp enc.c:425:
```

Fig. 5 Cifrado del archivo sin padding

4. Con el comando `ls -l` se listan los archivos para comprobar su tamaño (en bytes).

```
cripto@lab:~/aes$ ls -l
total 12
-rw-r--r-- 1 cripto cripto 112 oct  5 02:18 mensajeNoPadding.enc
-rw-r--r-- 1 cripto cripto 128 oct  5 02:03 mensajePadding.enc
-rwxr-xr-x 1 cripto cripto 118 oct  5 01:58 mensaje.txt
```

Fig. 6 Verificación del tamaño de los archivos

Se puede observar que el tamaño del archivo *mensajeNoPadding.enc* es menor en 10 bytes al archivo *mensajePadding.enc*, y 6 bytes menor al archivo sin cifrar *mensaje.txt*, esto se debe a que AES en el modo de operación ECB cifra de forma independiente bloques de 128 bits o 16 bytes, y los últimos 6 bytes, al no completar dicho bloque, no los cifra.

5. Con el comando *hexdump* se hará otra comprobación para analizar cómo es el cifrado por bloques.

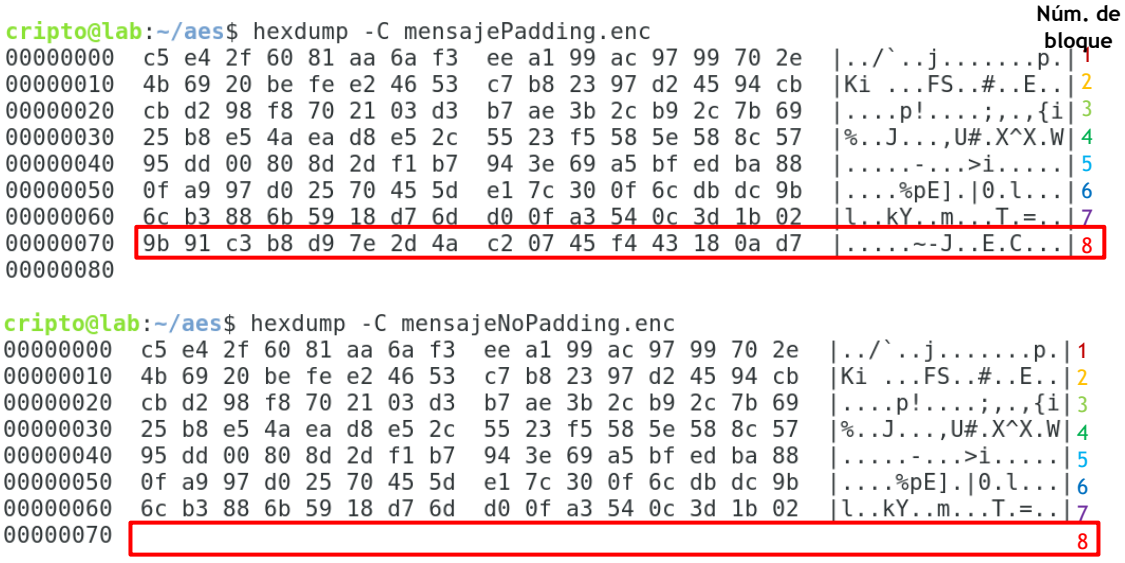

Fig. 7 Comparación de los bloques cifrados

Como puede observarse, el archivo cifrado *mensajePadding.enc* tiene ocho bloques de 16 bytes cada uno, mientras que *mensajeNoPadding.enc* solamente tiene siete bloques de 16 bytes, es decir, le falta un bloque y se presume que hubo pérdida de información.

6. Por último, descifre los dos archivos cifrados con los siguientes comandos:

```
$ openssl enc -d -v -aes-128-ecb -K 6e4b27a9a3a32165eb29eff45c204b0a -in \
mensajePadding.enc

$ openssl enc -d -v -aes-128-ecb -K 6e4b27a9a3a32165eb29eff45c204b0a -in \
mensajeNoPadding.enc
```

Observe que el contenido del archivo del mensaje en claro sí fue descifrado en su totalidad cuando se descifra el archivo *mensajePadding.enc*, mientras que cuando se descifra el archivo *mensajeNoPadding.enc*, el mensaje está incompleto, puesto que desde el proceso de cifrado no se cifró de forma completa.

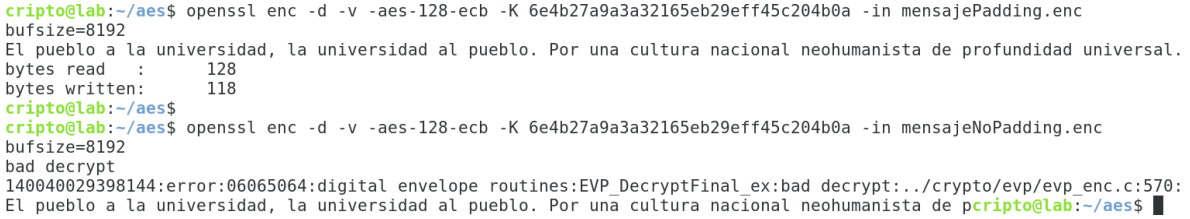

Fig. 8 Descifrado de los archivos

7. Pruebe a cifrar el mismo archivo, pero ahora sin indicar la llave en el comando (omita el argumento *-K* y la llave). Se le pedirá ingresar una contraseña.

8. Modifique el comando de tal forma que:

    a. No se despliegue la advertencia sobre la función de derivación de llave
    b. Se indique en la salida la llave empleada para el cifrado (es decir, la llave derivada de la contraseña que usted ingresó).

9. Modifique nuevamente el comando de tal forma que se cifre el archivo utilizando algún otro modo de operación.

PARTE 2: Poniendo a prueba AES-ECB

Introducción

ECB – Electronic Code Book

Cada bloque del mensaje en claro es cifrado de forma independiente usando la misma la llave. Un bloque del mensaje en claro siempre produce el mismo criptograma para la misma llave. Si se presenta un error durante el cifrado, solamente afectará a un bloque de cifrado.

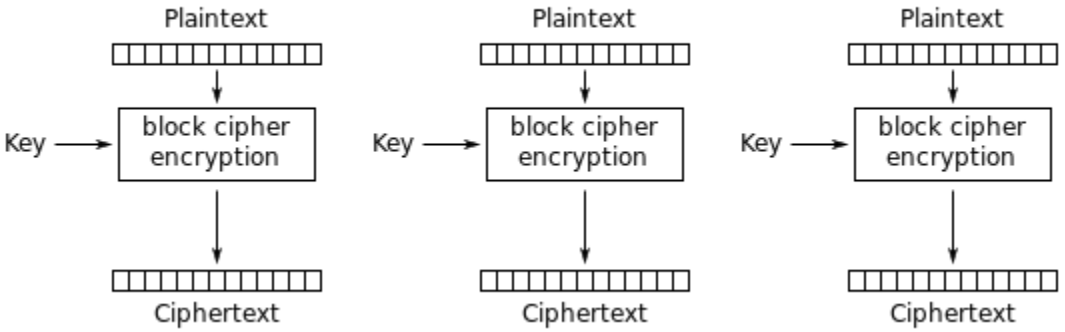

Fig. 9 Cifrado con ECB

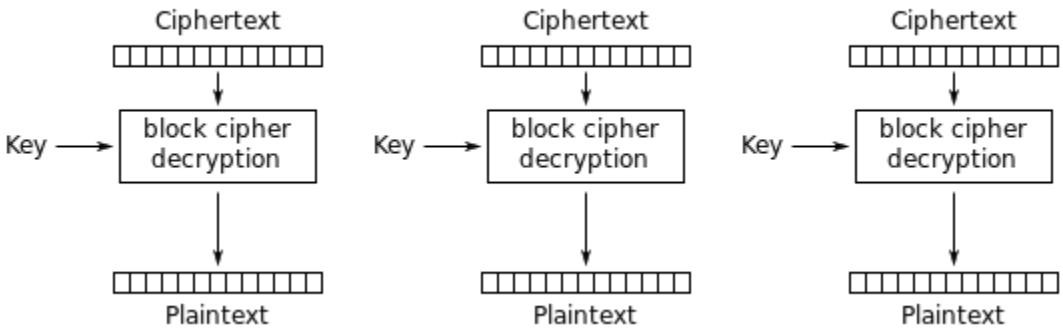

Fig. 10 Descifrado con ECB

CBC – Cipher Block Chain

Como primer paso se realiza una operación XOR entre el vector de inicialización (IV) y el mensaje en claro, antes de entrar al algoritmo de cifrado. Después se repite el proceso, antes de entrar al algoritmo de cifrado, a cada bloque de texto se le aplica la operación XOR con el criptograma correspondiente al bloque anterior. Cada bloque cifrado depende de los criptogramas anteriores. Un bloque del mensaje en claro siempre produce el mismo bloque cifrado para la misma llave y para el mismo IV. Diferentes IV previenen que se tenga el mismo criptograma para el mismo mensaje en claro. Un error en el cifrado afecta al bloque cifrado actual y a todos los siguientes.

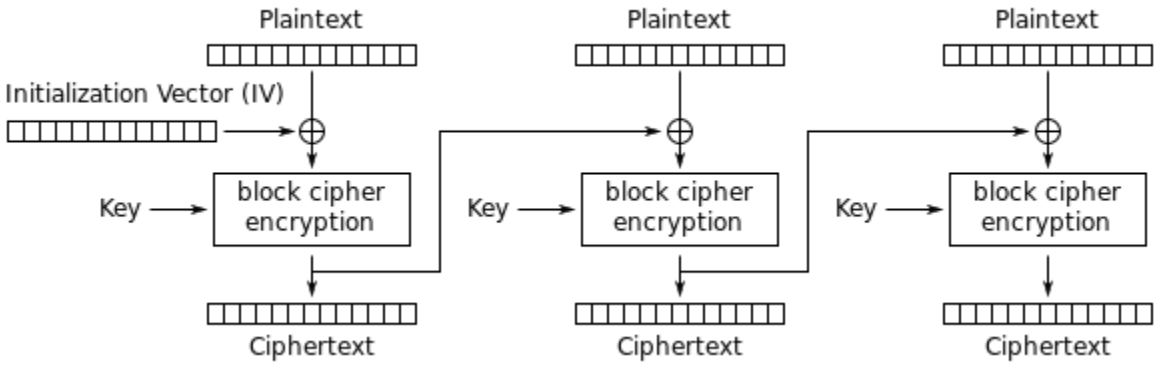

Fig. 11 Cifrado con CBC

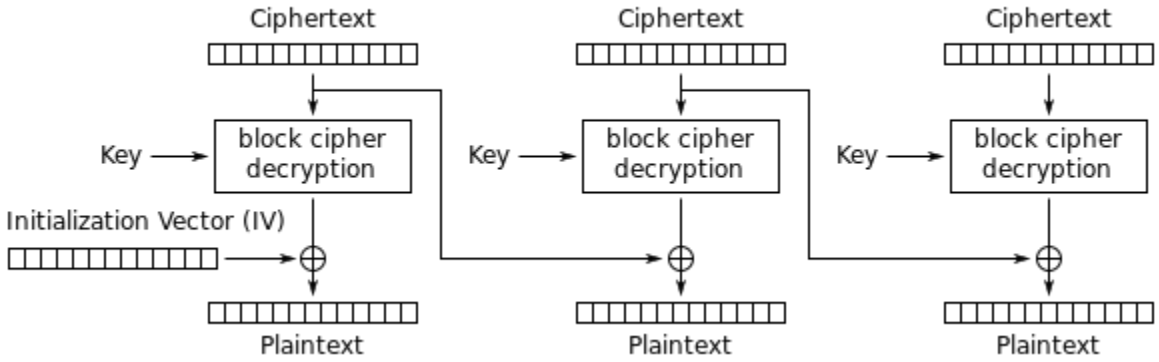

Fig. 12 Descifrado con CBC

Formato de imagen ppm

El formato de imagen ppm es un formato adecuado para realizar prácticas de procesamiento digital de imágenes, porque es un formato sin compresión y es posible almacenarlo en formato de texto, por lo que no es necesario realizar un tratamiento adicional del archivo donde se almacena la imagen. Además, existen múltiples herramientas de conversión entre los formatos más comunes de imágenes y el formato ppm.

Desarrollo

1. Una imagen en formato ppm fue cifrada usando el algoritmo AES con una llave de 128 bits. Se cifró de dos formas diferentes: una usando el modo de operación ECB y la otra usando el modo de operación CBC.

2. Obtenga una imagen que le de indicios de la original, a partir de los archivos proporcionados[2] que se listan a continuación:

    - `data`
      Cabecera sin cifrar de la imagen original.

    - `split_image-ecb.encrypted`
      Imagen cifrada con AES-128 en modo ECB, al archivo se le retiró la cabecera cifrada.

    - `full_image-ecb.encrypted`
      Imagen cifrada con AES-128 en modo ECB.

    - `split_image-cbc.encrypted`
      Imagen cifrada con AES-128 en modo CBC, al archivo se le retiró la cabecera cifrada.

    - `full_image-cbc.encrypted`
      Imagen cifrada con AES-128 en modo CBC.

Nota importante: Los archivos que le corresponden son los de la carpeta con el número obtenido al ejecutar el siguiente comando:

```
$ python -c "print(int(''.join(format(ord(char), 'x') for char in 'NOMBRE APELLIDO'),16) % 15)"
```

Es importante que coloque, en la parte resaltada en color verde, su primer nombre y apellido. **Añada una captura de esto en su reporte.**

[2] Usted debe determinar cuáles archivos le serán útiles y cuáles no.

Deberá obtener dos imágenes, una en donde se observe de forma parcial la imagen original, y la otra en donde no se distinga la imagen original. Se colocan las siguientes figuras a modo de ejemplo:

Fig. 13 Imagen cifrada con AES

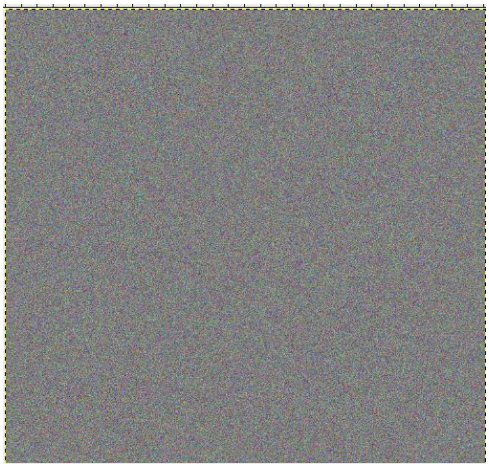

Fig. 6 Imagen cifrada con AES

Cuestionario

1. ¿De cuántos bits es la llave `6e4b27a9a3a32165eb29eff45c204b0a` que se usó tanto para cifrar como para descifrar los archivos? Justifique su respuesta.
2. Ejecute el siguiente comando
   `openssl enc -v -aes-128-ecb -K` **`64f3ad6017h65eb2f5f4e0b82f5ceb31`** `-in \ mensaje.txt > mensajeCifrado.enc`
   Tome en cuenta que la llave cambió. ¿Se cifró el archivo?, ¿qué error indica OpenSSL?, ¿a qué le atribuye la causa de ese error?
3. ¿Cuál es el riesgo de introducir la llave en el mismo comando?
4. ¿Qué opciones de OpenSSL necesitó indicar para evitar la advertencia sobre la derivación de llave y para desplegar la llave derivada de la contraseña que usted ingresó? ¿Qué otra información se despliega?
5. ¿Cuál es la opción en el comando de OpenSSL que se usa para indicar que se descifrará el archivo?
6. Investigue las diferentes técnicas de *padding* para el cifrado de archivos.
7. ¿En qué modo de operación sí se puede obtener (parcialmente) la imagen, y por qué?

Elementos a calificar

1. Redacte un reporte en el que indique los pasos que considere necesarios para explicar cómo realizó la práctica, incluyendo capturas de pantalla que justifiquen su trabajo.
2. Incluya en su reporte tanto las respuestas del Cuestionario, como un apartado de conclusiones referentes al trabajo realizado.
3. Puede agregar posibles errores, complicaciones, opiniones, críticas de la práctica o del laboratorio, o cualquier comentario relativo a la misma.
4. Deberá subir el reporte en PDF a Classroom.

Referencias

- The ECB Penguin. https://words.filippo.io/the-ecb-penguin/
- OpenSSL. *Manual de OpenSSL*. https://wiki.openssl.org/index.php/Enc
- NIST Federal Information Processing Standars Publications. *FIPS 197. Advanced Encryption Standard - AES*, 2001.
- W. Stallings. *Cryptography and Network Security. Principles and Practice*. Pearson. 2014.

Universidad Nacional Autónoma de México

Paulo Contreras Flores
paulo.contreras.flores@ciencias.unam.mx

Tonatihu Sánchez Neri
tonatihus@ciencias.unam.mx