#### Marco Manuel Fernánez Pranno - SPSI 2016

#### Repositorio de la práctica

### Tareas a realizar

- 1. Crear archivo binario de 1024 bits, todo a 0. (input.bin)
- 2. Crear archivo binario de 1024 bits, un bit a 1 entre los primeros 40, el resto todo a 0. (input1.bin)
- 3. Cifrar input.bin con DES en modo ECB, CBC y 0FB; usando una clave débil y otra semidébil. (Cualquier vector de inicialización)
  - 3.1. DES con clave débil, modos ECB, CBC y OFB:

```
openssl des-ecb -nosalt -in input.bin -out input0-des-ecb-debil.enc
-K 0000000
openssl des-cbc -nosalt -in input.bin -out input0-des-cbc-debil.enc
-iv 00000000 -K 0000000
openssl des-ofb -nosalt -in input.bin -out input0-des-ofb-debil.enc
-iv 00000000 -K 0000000
```

```
marcofp@debian:~/Dropbox/ETSIIT/Ultimo/SPSI/P2$ openssl des-ofb -nosalt -in input.bin -out input0-des-ofb-debil.enc -iv 000000000 -K 00000000
marcofp@debian:~/Dropbox/ETSIIT/Ultimo/SPSI/P2$ openssl des-cbc -nosalt -in input.bin -out input0-des-cbc-debil.enc -iv 000000000 -K 00000000
marcofp@debian:~/Dropbox/ETSIIT/Ultimo/SPSI/P2$ openssl des-ecb -nosalt -in input.bin -out input0-des-ecb-debil.enc -K 0000000
marcofp@debian:~/Dropbox/ETSIIT/Ultimo/SPSI/P2$ ls *.enc
input0-des-cbc-debil.enc input0-des-ecb-debil.enc input0-des-ofb-debil.enc
marcofp@debian:~/Dropbox/ETSIIT/Ultimo/SPSI/P2$
```

• 3.2. DES con clave semi-débil, modos ECB, CBC y OFB:

```
openssl des-ecb -nosalt -in input.bin -out input0-des-ecb-
semidebil.enc -K 01FE
openssl des-cbc -nosalt -in input.bin -out input0-des-cbc-
semidebil.enc -iv 000000000 -K 01FE
openssl des-ofb -nosalt -in input.bin -out input0-des-ofb-
semidebil.enc -iv 000000000 -K 01FE
```

```
marcofp@debian:~/Dropbox/ETSIIT/Ultimo/SPSI/P2$ openssl des-ecb -nosalt -in input.bin -out input0-des-ecb-semidebil.enc -K 01FE
marcofp@debian:~/Dropbox/ETSIIT/Ultimo/SPSI/P2$ openssl des-cbc -nosalt -in input.bin -out input0-des-cbc-semidebil.enc -iv 000000000 -K 01FE
marcofp@debian:~/Dropbox/ETSIIT/Ultimo/SPSI/P2$ openssl des-ofb -nosalt -in input.bin -out input0-des-ofb-semidebil.enc -iv 00000000 -K 01FE
marcofp@debian:~/Dropbox/ETSIIT/Ultimo/SPSI/P2$ ls *-semidebil.enc
input0-des-cbc-semidebil.enc input0-des-ecb-semidebil.enc
marcofp@debian:~/Dropbox/ETSIIT/Ultimo/SPSI/P2$ []
```

# 4. Cifrar input.bin e input1.bin con DES en modo ECB y clave a elegir. Explicar la forma de los resultados obtenidos.

• 4.1. Input.bin DES modo ECB (clave semidébil):

```
openssl des-ecb -nosalt -in input.bin -out input0-des-ecb-
semidebil.enc -K 01FE
```

• 4.2. Input1.bin DES modo ECB (clave semidébil):

```
openssl des-ecb -nosalt -in input1.bin -out input1-des-ecb-
semidebil.enc -K 01FE
```

```
marcofp@debian:~/Dropbox/ETSIIT/Ultimo/SPSI/P2$ openssl des-ecb -nosalt -in input.bin -out input0-des-ecb-semidebil.enc -K 01FE
marcofp@debian:~/Dropbox/ETSIIT/Ultimo/SPSI/P2$ openssl des-ecb -nosalt -in input1.bin -out input1-des-ecb-semidebil.enc -K 01FE
marcofp@debian:~/Dropbox/ETSIIT/Ultimo/SPSI/P2$ ls *.enc
input0-des-ecb-semidebil.enc input1-des-ecb-semidebil.enc
marcofp@debian:~/Dropbox/ETSIIT/Ultimo/SPSI/P2$
```

#### • 4.3. Comparación de resultados:

Vemos en ambos casos, debido a la naturaleza del algoritmo de cifrado, que se repite un patrón dentro del archivo cifrado; en ambos casos ocurre lo mismo pero con un patrón diferente que difiere en un sólo caso, el bit de diferencia entre ellos.

```
File Edit View Windows Help
0000000012 38 B9 D0 03 36 98 A9 AA 6D 99 45 EC 83 72 B3
00000010AA 6D 99 45 EC 83 72 B3 AA 6D 99 45 EC 83 72 B3
00000020AA 6D 99 45 EC 83 72 B3 AA 6D 99 45 EC 83 72 B3
00000030AA 6D 99 45 EC 83 72 B3 AA 6D 99 45 EC 83 72 B3
00000040AA 6D 99 45 EC 83 72 B3 AA 6D 99 45 EC 83 72 B3
00000050AA 6D 99 45 EC 83 72 B3 AA 6D 99 45 EC 83 72 B3
00000060AA 6D 99 45 EC 83 72 B3 AA 6D 99 45 EC 83 72 B3
00000070AA 6D 99 45 EC 83 72 B3 AA 6D 99 45 EC 83 72 B3
0000008009 2C 3F 7B D0 ED E7 ED
Ø - +
                          input0-des-ecb-semidebil.enc - GHex
File Edit View Windows Help
00000000AA 6D 99 45 EC 83 72 B3 AA 6D 99 45 EC 83 72 B3
00000010AA 6D 99 45 EC 83 72 B3 AA 6D 99 45 EC 83 72 B3
00000020AA 6D 99 45 EC 83 72 B3 AA 6D 99 45 EC 83 72 B3
00000030AA 6D 99 45 EC 83 72 B3 AA 6D 99 45 EC 83 72 B3
00000040AA 6D 99 45 EC 83 72 B3 AA 6D 99 45 EC 83 72 B3
00000050AA 6D 99 45 EC 83 72 B3 AA 6D 99 45 EC 83 72 B3
00000060AA 6D 99 45 EC 83 72 B3 AA 6D 99 45 EC 83 72 B3
00000070AA 6D 99 45 EC 83 72 B3 AA 6D 99 45 EC 83 72 B3
0000008009 2C 3F 7B D0 ED E7 ED
```

- 5. Cifrar input.bin e input1.bin con DES en modo CBC, clave y vector de inicialización a elegir. Comparar resultados con el apartado anterior.
  - 5.1. Input.bin DES modo CBC (clave débil, todo 1), vector de inicialización todo 0:

```
openssl des-cbc -nosalt -in input.bin -out input0-des-cbc-
semidebil.enc -iv 00000000 -K 11111111
```

• 5.2. Input1.bin DES modo CBC (clave débil, todo 1), vector de inicialización todo 0:

```
openssl des-cbc -nosalt -in input1.bin -out input1-des-cbc-
```

```
marcofp@debian:~/Dropbox/ETSIIT/Ultimo/SPSI/P2$ openssl des-cbc -nosalt -in input.bin -out input0-des-cbc-semidebil.enc -iv 000000000 -K 11111
marcofp@debian:~/Dropbox/ETSIIT/Ultimo/SPSI/P2$ openssl des-cbc -nosalt -in input1.bin -out input1-des-cbc-semidebil.enc -iv 000000000 -K 1111
1111
marcofp@debian:~/Dropbox/ETSIIT/Ultimo/SPSI/P2$ ls *.enc
input0-des-cbc-semidebil.enc input1-des-cbc-semidebil.enc
marcofp@debian:~/Dropbox/ETSIIT/Ultimo/SPSI/P2$ |
```

#### • 5.3. Comparación de resultados:

Vemos como gracias a la aplicación de un vector de incialización y la diferencia del algoritmo CBC frente a ECB los patrones evidentes del apartado anterior en el texto cifrado desaparecen y no se aprencia rastro de ninguno. Deducimos de ésto que ECB es un cifrado más sencillo (más débil), frente a CBC que ofusca de una forma más efectiva el contenido del archivo.

```
File Edit View Windows Help
00000000C3 A8 C5 6D E8 15 EB 61 7E C2 16 8F 8D A9 14 A8...m...a~.....
0000001040 75 77 35 44 50 0E 11 ED D2 42 24 3F 6E EF 84@uw5DP....B$?n..
0000002034 3E 7C F5 0D E6 96 69 EC 9F BB D1 52 3C 63 704>|...i...R<cp
00000030EF FF 70 E6 9F 94 5E EE 3D 70 F3 A3 C3 B4 D8 0F..p...^.=p.....
0000004086 34 39 51 1C 49 A3 91 4A 57 D6 84 00 61 9E 22.49Q.I..JW...a."
000000506F 53 E9 82 A3 97 0F 7D 59 20 23 71 F7 B2 6C 0EoS.....}Y #q..l.
0000007045 FD E3 51 DB D9 D1 AF E9 BB 7F B6 E3 C2 C1 2DE..O......
0000008000 36 A4 2F 9F 52 01 75
                                                      .6./.R.u
⊗ - +
                        input1-des-cbc-semidebil.enc - GHex
File Edit View Windows Help
00000000007 53 45 F4 77 17 C3 8E B9 1F 10 70 05 BC 1F 93.SE.w.....p....
000000101D 7F 24 D8 82 98 A0 7A 04 1D F3 26 03 7B F5 35..$...z...&.{.5
00000020DA 61 A4 99 DC 2F 67 77 47 EF E8 77 7F A8 DB D4.a.../gwG..w....
000000300F F1 C9 D2 9E 54 7A D8 FB B1 FA 8F 48 90 D7 AE.....Tz.....H...
000000404A 4F 65 DF D3 77 18 FC 3E 8E D0 F3 82 46 65 91J0e..w..>....Fe.
00000050B8 44 87 B8 22 6B AD 31 A1 79 7F 1D 4A 7F 87 C5.D.."k.1.y..J...
0000006011 1E 5B DF 51 02 7F 49 2A EE DA 8C 73 B8 25 6F..[.0..I*...s.%o
000000700C D3 A8 D5 C4 1E A8 22 CD C5 2B 40 10 40 95 54...........+a.a.T
000000805D E4 A5 38 C8 55 09 86
                                                      1..8.U..
```

### 6. Repetir los puntos 3 a 5 con AES-128 y AES-256.

## (3) Cifrar input.bin con AES-128 y AES-256 en modo ECB, CBC y 0FB; usando una clave débil y otra semidébil. (Cualquier vector de inicialización)

#### • 6.1. AES-128:

#### • 6.1.1. Modo ECB:

Clave débil:

openssl aes-128-ecb -nosalt -in input.bin -out input0aes128-ecb-debil.enc -K 11111111

Clave semidébil:

openssl aes-128-ecb -nosalt -in input.bin -out input0aes128-ecb-semidebil.enc -K 01FE

#### • 6.1.2. Modo CBC:

Clave débil:

openssl aes-128-cbc -nosalt -in input.bin -out input0aes128-cbc-debil.enc -iv 00000000 -K 11111111

Clave semidébil:

openssl aes-128-cbc -nosalt -in input.bin -out input0-aes128-cbc-semidebil.enc -iv 00000000 -K 01FE

#### • 6.1.3. Modo OFB:

Clave débil:

openssl aes-128-ofb -nosalt -in input.bin -out input0aes128-ofb-debil.enc -iv 00000000 -K 11111111

Clave semidébil:

openssl aes-128-ofb -nosalt -in input.bin -out input0aes128-ofb-semidebil.enc -iv 00000000 -K 01FE

#### • 6.2. AES-256:

#### • 6.2.1. Modo ECB:

#### Clave débil:

openssl aes-256-ecb -nosalt -in input.bin -out input0aes256-ecb-debil.enc -K 11111111

Clave semidébil:

openssl aes-256-ecb -nosalt -in input.bin -out input0aes256-ecb-semidebil.enc -K 01FE

#### 6.2.2. Modo CBC:

Clave débil:

openssl aes-256-cbc -nosalt -in input.bin -out input0aes256-cbc-debil.enc -iv 00000000 -K 11111111

Clave semidébil:

openssl aes-256-cbc -nosalt -in input.bin -out input0aes256-cbc-semidebil.enc -iv 00000000 -K 01FE

#### • 6.2.3. Modo OFB:

Clave débil:

openssl aes-256-ofb -nosalt -in input.bin -out input0-aes256-ofb-debil.enc -iv 00000000 -K 11111111

Clave semidébil:

openssl aes-256-ofb -nosalt -in input.bin -out input0aes256-ofb-semidebil.enc -iv 00000000 -K 01FE

```
marcofp@debian:~/Dropbox/ETSIIT/Ultimo/SPSI/P2$ openssl aes-128-ecb -nosalt -in input.bin -out input0-aes128-ecb-debil.enc -K 11111111
marcofp@debian:~/Dropbox/ETSIIT/Ultimo/SPSI/P2$ openssl aes-128-ecb -nosalt -in input.bin -out input0-aes128-ecb-semidebil.enc -K 01FE
marcofp@debian:~/Dropbox/ETSIIT/Ultimo/SPSI/P2$ openssl aes-128-cbc -nosalt -in input.bin -out input0-aes128-cbc-debil.enc -iv 00000000 -K 11
11111
marcofp@debian:~/Dropbox/ETSIIT/Ultimo/SPSI/P2$ openssl aes-128-cbc -nosalt -in input.bin -out input0-aes128-cbc-semidebil.enc -iv 00000000 - K 01FE
marcofp@debian:~/Dropbox/ETSIIT/Ultimo/SPSI/P2$ openssl aes-128-ofb -nosalt -in input.bin -out input0-aes128-ofb-debil.enc -iv 00000000 -K 11
11111
marcofp@debian:~/Dropbox/ETSIIT/Ultimo/SPSI/P2$ openssl aes-128-ofb -nosalt -in input.bin -out input0-aes128-ofb-semidebil.enc -iv 00000000 -K 01FE
marcofp@debian:~/Dropbox/ETSIIT/Ultimo/SPSI/P2$ ls *.enc
input0-aes128-cbc-debil.enc input0-aes128-ecb-debil.enc input0-aes128-ofb-semidebil.enc
input0-aes128-cbc-semidebil.enc input0-aes128-ecb-semidebil.enc input0-aes128-ofb-semidebil.enc
marcofp@debian:~/Dropbox/ETSIIT/Ultimo/SPSI/P2$ ls *.enc
input0-aes128-cbc-semidebil.enc input0-aes128-ecb-semidebil.enc input0-aes128-ofb-semidebil.enc
marcofp@debian:~/Dropbox/ETSIIT/Ultimo/SPSI/P2$
```

(4) Cifrar input.bin e input1.bin con AES-128 y AES-256 en modo ECB y clave a elegir. Explicar la forma de los resultados obtenidos.

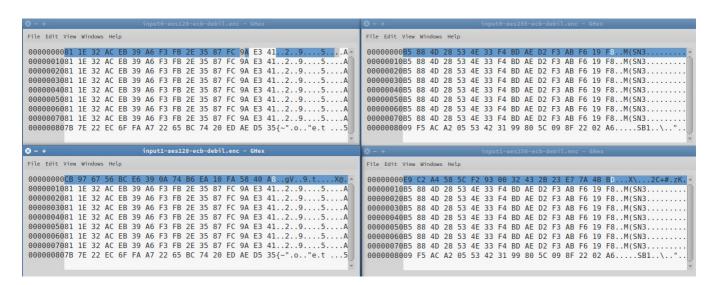
openssl aes-128-ecb -nosalt -in input.bin -out input0-aes128-ecb-debil.enc -K 1111111 openssl aes-256-ecb -nosalt -in input.bin -out

input0-aes256-ecb-debil.enc -K 11111111 openssl aes-128-ecb -nosalt -in
input1.bin -out input1-aes128-ecb-debil.enc -K 11111111 openssl aes256-ecb -nosalt -in input1.bin -out input1-aes256-ecb-debil.enc -K
11111111

```
marcofp@debian:~/Dropbox/ETSIIT/Ultimo/SPSI/P2$ openssl aes-128-ecb -nosalt -in input.bin -out input0-aes128-ecb-debil.enc -K 11111111
marcofp@debian:~/Dropbox/ETSIIT/Ultimo/SPSI/P2$ openssl aes-256-ecb -nosalt -in input.bin -out input0-aes256-ecb-debil.enc -K 11111111
marcofp@debian:~/Dropbox/ETSIIT/Ultimo/SPSI/P2$ openssl aes-128-ecb -nosalt -in input1.bin -out input1-aes128-ecb-debil.enc -K 11111111
marcofp@debian:~/Dropbox/ETSIIT/Ultimo/SPSI/P2$ openssl aes-256-ecb -nosalt -in input1.bin -out input1-aes256-ecb-debil.enc -K 11111111
marcofp@debian:~/Dropbox/ETSIIT/Ultimo/SPSI/P2$ ls *.enc
input0-aes128-ecb-debil.enc input0-aes256-ecb-debil.enc input1-aes128-ecb-debil.enc input1-aes256-ecb-debil.enc
marcofp@debian:~/Dropbox/ETSIIT/Ultimo/SPSI/P2$
```

#### • Comparación:

Igual que en el caso del ejercicio 4 con DES, vemos como se repiten patrones en el texto cifrado, variando estos en cada fichero.



## (5) Cifrar input.bin e input1.bin con AES-128 y AES-256 en modo CBC, clave y vector de inicialización a elegir. Comparar resultados con el apartado anterior.

```
openssl aes-128-cbc -nosalt -in input.bin -out input0-aes128-cbc-debil.enc -iv 00000000 -K 11111111

openssl aes-256-cbc -nosalt -in input.bin -out input0-aes256-cbc-debil.enc -iv 00000000 -K 11111111

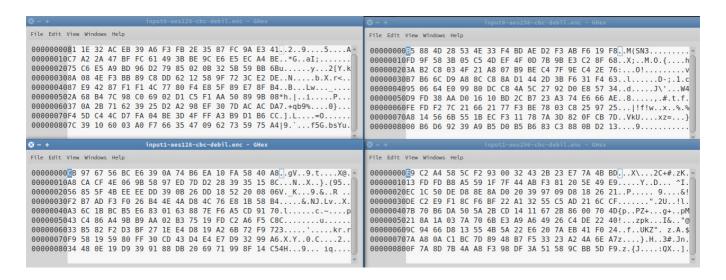
openssl aes-128-cbc -nosalt -in input1.bin -out input1-aes128-cbc-debil.enc -iv 00000000 -K 11111111

openssl aes-256-cbc -nosalt -in input1.bin -out input1-aes256-cbc-debil.enc -iv 00000000 -K 11111111
```

```
marcofp@debian:~/Dropbox/ETSIIT/Ultimo/SPSI/P2$ openssl aes-128-cbc -nosalt -in input.bin -out input0-aes128-cbc-debil.enc -iv 000000000 -K 11
111111
marcofp@debian:~/Dropbox/ETSIIT/Ultimo/SPSI/P2$ openssl aes-256-cbc -nosalt -in input.bin -out input0-aes256-cbc-debil.enc -iv 000000000 -K 11
111111
marcofp@debian:~/Dropbox/ETSIIT/Ultimo/SPSI/P2$ openssl aes-128-cbc -nosalt -in input1.bin -out input1-aes128-cbc-debil.enc -iv 00000000 -K 1
1111111
marcofp@debian:~/Dropbox/ETSIIT/Ultimo/SPSI/P2$ openssl aes-256-cbc -nosalt -in input1.bin -out input1-aes256-cbc-debil.enc -iv 00000000 -K 1
1111111
marcofp@debian:~/Dropbox/ETSIIT/Ultimo/SPSI/P2$ ls *.enc
input0-aes128-cbc-debil.enc input0-aes256-cbc-debil.enc input1-aes128-cbc-debil.enc input1-aes256-cbc-debil.enc
marcofp@debian:~/Dropbox/ETSIIT/Ultimo/SPSI/P2$ ]
```

#### • Comparación:

De nuevo, igual que en el caso del ejercicio 5 con DES, los patrones en el texto cifrado desaparecen al aplicar el algoritmo CBC frente al ECB, haciendo uso de un vector de inicialización y gracias a las operaciones XOR.



# 7. Cifrar input.bin con AES-192 en modo 0FB, clave y vector de inicialización a elegir. (Salida output.bin)

```
openssl aes-192-ofb -nosalt -in input.bin -out output.bin -iv 00000000
-K 01FE
```

```
marcofp@debian:~/Dropbox/ETSIIT/Ultimo/SPSI/P2$ openssl aes-192-ofb -nosalt -in input.bin -out output.bin -iv 000000000 -K 01FE
marcofp@debian:~/Dropbox/ETSIIT/Ultimo/SPSI/P2$ ls *.bin
input1.bin input.bin output.bin
marcofp@debian:~/Dropbox/ETSIIT/Ultimo/SPSI/P2$ [
```

## 8. Descrifrar output.bin utilizando la misma clave y vector de inicialización que en 7.

openssl aes-192-ofb -nosalt -d -in output.bin -out output-

```
marcofp@debian:~/Dropbox/ETSIIT/Ultimo/SPSI/P2$ openssl aes-192-ofb -nosalt -d -in output.bin -out output-descrifrado.bin -iv 000000000 -K 01F
E
marcofp@debian:~/Dropbox/ETSIIT/Ultimo/SPSI/P2$ ls output*
output.bin output-descrifrado.bin
marcofp@debian:~/Dropbox/ETSIIT/Ultimo/SPSI/P2$ [
```

Vemos como el contenido de output-descrifrado.bin es igual a input.bin:

```
output-descrifrado.bin - GHex
Ø - +
File Edit View Windows Help
0000001000 00 00
                           00 00 00 00
                                      00
                                           00
              00 00 00
                      00
                         00
                                        00
                                             00......
0000002000 00 00
                                 00 00
                                      00
                                           00
                                             00.....
               00 00 00
                      00
                         00
                           00
                              00
                                        00
0000003000 00 00
              00 00 00
                      00
                         00
                           00
                              00
                                 00
                                   00
                                      00
                                        00
                                          00 00.....
0000004000 00 00
               00 00 00
                      00
                         00
                           00
                              00
                                 00
                                   00
                                      00
                                        00
                                           00 00.....
0000005000 00 00
               00 00 00
                      00
                         00
                           00
                              00
                                 00
                                   00
                                      00
                                        00
                                           00
                                             00.....
0000006000 00 00
               00 00 00
                      00 00 00 00 00 00
                                      00
                                        00
                                           00 00..
00 00 00.
```

9. Volver a cifrar output.bin con AES-192 en modo 0FB, clave y vector de inicialización del punto 7. Comparar el resultado con lo obtenido en el punto 8.

```
openssl aes-192-ofb -nosalt -in output.bin -out output-cifrado-
doble.bin -iv 00000000 -K 01FE
```

```
arcofp@debian:~/Dropbox/ETSIIT/Ultimo/SPSI/P2$ ls output
utput-cifrado-doble.bin output-descrifrado.bin
narcofp@debian:~/Dropbox/ETSIIT/Ultimo/SPSI/P2$
00000000<mark>A</mark>5 24 C7 37 F4 2B 4D A9 25 D7 21 10
                                                                  00
00000000065 24 C7 37 F4 28 40 A9 23 07 21 10 72 31 00 00000000000 95 56 00 37 48 D6 A1 E9 77 00 2F 9E 7A 6A D1 2B.V.7K...../.zj.+
00000002006 A6 C7 38 CA 62 BA 21 25 9D F2 DE ED 72 32 A0...8.b.1%....r2.
                                                                 00.
                                                                  00000040F7 9E 08 FD 56 8C 48 38 6A F3 6A 67 56 F4 2C 26....V.H8j.jgV.,&
000000507C B1 F8 18 09 E4 F0 2D C0 C5 D8 5D 32 07 33 BF|......]2.3.
                                                                  0000006013 C6 15 7E 70 97 D2 38 45 47 CD 0C 7B 5D 2D B9...~p..8EG.
                                                                  00000070B2 42 5F BE C2 F4 0F 47 64 85 45 B6 30 2A E5 53.B ....Gd.E.0
```

- 10. Presentar la descripción de otro cifrado simétrico que aparezca en la implementación de OpenSSL.
- RC4: RC4 es un sistema de cifrado de flujo muy utilizado, forma parte de

protocolos tan populares como TLS/SSL, utilizado para cifrar el tráfico en las conexiones por internet; y también para el protocolo WEP en redes inalámbricas.

RC4 ha sido excluído de los estándares de alta seguridad ya que se conocen formas de vulnerarlo fácilmente. Por ese motivo es muy poco recomendable emplear WEP para cifrado de redes inalámbricas.

RC4 genera flujos pseudoaleatórios de bits (keystreams), para cifrar un texto éste se combina con los keystreams utilizando una operación XOR.

## 11. Repetir puntos de 3 a 5 con el cifrado presentado en el punto anterior.

Debido a que RC4 no dispone de modos ECB, CBC y OFB, para este apartado se utilizará el cifrado RC2, diseñado por el mismo creador de RC4. A diferencia de RC4, que es un cifrado de flujo, RC2 es un cifrado de bloque; aunque también es simétrico por lo tanto nos sirve para el experimento.

### (3) Cifrar input.bin con RC4 en modo ECB, CBC y 0FB; usando una clave débil y otra semidébil. (Cualquier vector de inicialización)

#### • 11.1. RC2:

#### • 11.1.1. Modo ECB:

Clave débil:

```
openssl rc2-ecb -nosalt -in input.bin -out input0-rc2-ecb-debil.enc -K 11111111
```

Clave semidébil:

```
openssl rc2-ecb -nosalt -in input.bin -out input0-rc2-ecb-
semidebil.enc -K 01FE
```

#### • 11.1.2. Modo CBC:

Clave débil:

```
openssl rc2-cbc -nosalt -in input.bin -out input0-rc2-cbc-
```

debil.enc -iv 00000000 -K 11111111

Clave semidébil:

openssl rc2-cbc -nosalt -in input.bin -out input0-rc2-cbcsemidebil.enc -iv 00000000 -K 01FE

#### • 11.1.3. Modo OFB:

Clave débil:

openssl rc2-ofb -nosalt -in input.bin -out input0-rc2-ofb-debil.enc -iv 00000000 -K 11111111

Clave semidébil:

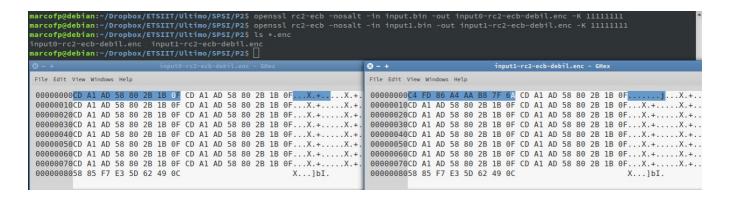
openssl rc2-ofb -nosalt -in input.bin -out input0-rc2-ofbsemidebil.enc -iv 00000000 -K 01FE

```
marcofp@debian:-/Dropbox/ETSIIT/Ultimo/SPSI/P2$ openssl rc2-ecb -nosalt -in input.bin -out input0-rc2-ecb-debil.enc -K 1111111
marcofp@debian:-/Dropbox/ETSIIT/Ultimo/SPSI/P2$ openssl rc2-ecb -nosalt -in input.bin -out input0-rc2-ecb-semidebil.enc -K 01FE
marcofp@debian:-/Dropbox/ETSIIT/Ultimo/SPSI/P2$ openssl rc2-cbc -nosalt -in input.bin -out input0-rc2-cbc-debil.enc -iv 000000000 -K 11111111
marcofp@debian:-/Dropbox/ETSIIT/Ultimo/SPSI/P2$ openssl rc2-cbc -nosalt -in input.bin -out input0-rc2-cbc-semidebil.enc -iv 000000000 -K 01FE
marcofp@debian:-/Dropbox/ETSIIT/Ultimo/SPSI/P2$ openssl rc2-ofb -nosalt -in input.bin -out input0-rc2-ofb-debil.enc -iv 000000000 -K 01FE
marcofp@debian:-/Dropbox/ETSIIT/Ultimo/SPSI/P2$ openssl rc2-ofb -nosalt -in input.bin -out input0-rc2-ofb-semidebil.enc -iv 000000000 -K 01FE
marcofp@debian:-/Dropbox/ETSIIT/Ultimo/SPSI/P2$ b *.enc
input0-rc2-cbc-debil.enc input0-rc2-ecb-debil.enc input0-rc2-ofb-debil.enc
input0-rc2-cbc-semidebil.enc input0-rc2-ecb-semidebil.enc input0-rc2-ofb-semidebil.enc
marcofp@debian:-/Dropbox/ETSIIT/Ultimo/SPSI/P2$
```

### (4) Cifrar input.bin e input1.bin con RC2 en modo ECB y clave a elegir. Explicar la forma de los resultados obtenidos.

openssl rc2-ecb -nosalt -in input.bin -out input0-rc2-ecb-debil.enc -K 11111111

openssl rc2-ecb -nosalt -in input1.bin -out input1-rc2-ecb-debil.enc -K

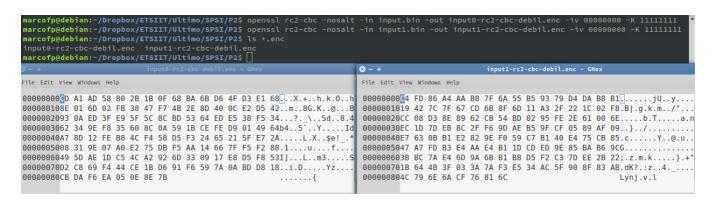


#### • Comparación:

Como se ha ido observando en la práctica en este tipo de ejercicio, ambos resultados presentan patrones que se repiten en el texto cifrado y difieren en una sección que corresponde al bit de diferencia entre el fichero input.bin y input1.bin.

## (5) Cifrar input.bin e input1.bin con RC2 en modo CBC, clave y vector de inicialización a elegir. Comparar resultados con el apartado anterior.

```
openssl rc2-cbc -nosalt -in input.bin -out input0-rc2-cbc-debil.enc -iv 00000000 -K 11111111 openssl rc2-cbc -nosalt -in input1.bin -out input1-rc2-cbc-debil.enc -iv 00000000 -K 11111111
```



#### • Comparación:

Como era predecible, los resultados no presentan patrones gracias a la diferencia entre los modos ECB y CBC, el uso de un vector de inicialización y la operación XOR.