

SEGURIDAD Y PROTECCIÓN DE SISTEMAS INFORMÁTICOS (2018-2019)  
GRADO EN INGENIERÍA INFORMÁTICA  
UNIVERSIDAD DE GRANADA

---

## PRACTICA 1

---



JUAN ALBERTO RIVERA PEÑA

## TAREAS

1. Partiremos de un archivo binario de 1024 bits, todos ellos con valor 0 . Para hacer referencia al mismo voy a suponer que se llama input.bin , pero podeis dar el nombre que os convenga.

Creamos el archivo input.bin de 128 bytes que es equivalente a 1024 bits con el comando dd:

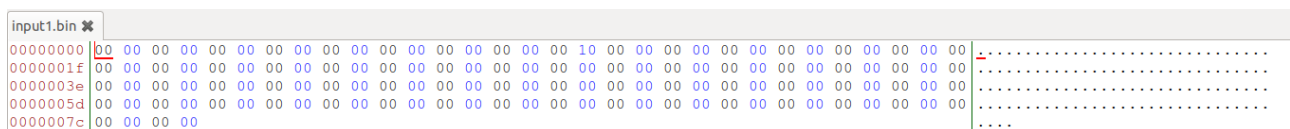
```
juan@juan-X541UAK:~/Documentos/Spsi/PRACTICAS/PRACTICA1$ dd if=/dev/zero of=input.bin bs=128 count=1
1+0 registros leídos
1+0 registros escritos
128 bytes copied, 0,000718174 s, 178 kB/s
```

2. Creamos otro archivo binario del mismo tamaño, que contenga un único bit con valor 1 entre los bits 130 y 150 , y todos los demás con valor 0 . Me referiré a este archivo como input1.bin

Creamos el archivo input1.bin de igual forma que en el ejercicio 1:

```
juan@juan-X541UAK:~/Documentos/Spsi/PRACTICAS/PRACTICA1$ dd if=/dev/zero of=input1.bin bs=128 count=1
1+0 registros leídos
1+0 registros escritos
128 bytes copied, 0,000967296 s, 132 kB/s
```

Y una vez creado lo modificamos con el editor 'Bless' para añadir el 1 entre los bits 130 y 150:



3. Cifrad input.bin e input1.bin con AES-256 en modos ECB , CBC y OFB usando una clave (no una contraseña) a elegir del tamaño adecuado, y con vector de inicialización 0123456789abcdef , cuando sea necesario. Explicad los diferentes resultados.

Ciframos el archivo input.bin de las tres maneras:

```
juan@juan-X541UAK:~/Documentos/Spsi/PRACTICAS/PRACTICA1$ openssl enc -aes-256-ecb -K 68546254526c324b62667874464f6f3 -in input.bin -out outputecb.bin
juan@juan-X541UAK:~/Documentos/Spsi/PRACTICAS/PRACTICA1$ openssl enc -aes-256-ofb -K 68546254526c324b62667874464f6f3 -in input.bin -out outputofb.bin -iv 0123456789abcdef
juan@juan-X541UAK:~/Documentos/Spsi/PRACTICAS/PRACTICA1$ openssl enc -aes-256-cbc -K 68546254526c324b62667874464f6f3 -in input.bin -out outputcbc.bin -iv 0123456789abcdef
```

```

juan@juan-X541UAK:~/Documentos/SpSI/PRACTICAS/PRACTICA1$ openssl enc -aes-256-ecb -K 68546254526c324b62667874464f6f3 -in input1.bin -o
ut output1ecb.bin
juan@juan-X541UAK:~/Documentos/SpSI/PRACTICAS/PRACTICA1$ openssl enc -aes-256-ofb -K 68546254526c324b62667874464f6f3 -in input1.bin -o
ut output1ofb.bin -iv 0123456789abcdef
juan@juan-X541UAK:~/Documentos/SpSI/PRACTICAS/PRACTICA1$ openssl enc -aes-256-cbc -K 68546254526c324b62667874464f6f3 -in input1.bin -o
ut output1cbc.bin -iv 0123456789abcdef

```

```

outputcb.bin
00000000 18 D4 20 3C F6 B3 E8 11 B2 12 21 F4 B5 8E BD 4A 18 D4 . <.....!...J..
00000012 20 3C F6 B3 E8 11 B2 12 21 F4 B5 8E BD 4A 18 D4 20 3C <.....!...J.. <
00000024 F6 B3 E8 11 B2 12 21 F4 B5 8E BD 4A 18 D4 20 3C F6 B3 .....!...J.. <..
00000036 E8 11 B2 12 21 F4 B5 8E BD 4A 18 D4 20 3C F6 B3 E8 11 .....!...J.. <...
00000048 B2 12 21 F4 B5 8E BD 4A 18 D4 20 3C F6 B3 E8 11 B2 12 !...!...J.. <.....
0000005a 21 F4 B5 8E BD 4A 18 D4 20 3C F6 B3 E8 11 B2 12 21 F4 !...J.. <.....!..
0000006c B5 8E BD 4A 18 D4 20 3C F6 B3 E8 11 B2 12 21 F4 B5 8E ...J.. <.....!...
0000007e BD 4A 01 6C 99 9B 2F 5F 1B EC EE 20 29 00 11 3E 17 C3 .J.1../_... )..>..
00000090

```

output1ecb.bin ✖		
00000000	18 D4 20 3C F6 B3 E8 11 B2 12 21 F4 B5 8E BD 4A 39 07	. <.....!....J9.
00000012	5C A9 1C B5 D9 76 A1 81 F0 F2 18 1F F7 F0 18 D4 20 3C	\....v....<
00000024	F6 B3 E8 11 B2 12 21 F4 B5 8E BD 4A 18 D4 20 3C F6 B3	.....!....J.<..
00000036	E8 11 B2 12 21 F4 B5 8E BD 4A 18 D4 20 3C F6 B3 E8 11	....!....J.<....
00000048	B2 12 21 F4 B5 8E BD 4A 18 D4 20 3C F6 B3 E8 11 B2 12	!!....J.<.....
0000005a	21 F4 B5 8E BD 4A 18 D4 20 3C F6 B3 E8 11 B2 12 21 F4	!....J.<.....!
0000006c	B5 8E BD 4A 18 D4 20 3C F6 B3 E8 11 B2 12 21 F4 B5 8E	....J.<.....!
0000007e	BD 4A 01 6C 99 9B 2F 5F 1B EC EE 20 29 00 11 3E 17 C3	.J.1../_... )..>..
00000090		

outputofb.bin	
00000000	CE 01 A9 C5 2F E3 F9 07 B8 5E 0C AB 43 9A 91 25 0B 0B . . . / . . . ^ . . C . . % . .
00000012	57 BC DC 29 00 11 D0 E7 9F 2B 46 38 8E 9F 91 77 E2 B8 W . . ) . . . . + F8 . . w . .
00000024	65 0C 97 45 AF 24 9D 9E B5 90 38 CF A5 F4 16 B4 2E 12 e . . E . \$ . . . . 8 . . . . .
00000036	44 27 50 0C E2 52 4F 19 9E 9D 92 31 99 D4 3C 75 80 D6 D ' P . . RO . . . . 1 . . < u . .
00000048	57 16 E7 F0 BC 1D 60 B0 D1 F9 75 14 E6 26 42 68 EE F1 W . . . . ` . . . u . . & B h . .
0000005a	47 F5 4A FB FB A6 FD 2C B3 75 11 B4 00 23 E2 40 76 F0 G . J . . . . , . u . . # . @ v .
0000006c	F1 59 E5 3D 19 4F 44 93 75 B3 E5 C1 52 C9 50 0F 36 F1 . Y . = . OD . u . . R . P . 6 .
0000007e	51 03 Q .

## Output1OFB:

output1ofb.bin																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																											
00000000	CE	01	A9	C5	2F	E3	F9	07	B8	5E	0C	AB	43	9A	91	25	0B	1B																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																									

En este algoritmo podemos ver que todos los bloques son iguales ya que no dependen unos bloques de otros excepto el que tiene un 1 que ese si cambia.

## OutputCBC:

outputcbc.bin																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
00000000	CE	01	A9	C5	2F	E3	F9	07	B8	5E	0C	AB	43	9A	91	25	0B	0B																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																										</

## Output1CBC:

output1cbc.bin ✖																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																											
00000000	CE	01	A9	C5	2F	E3	F9	07	B8	5E	0C	AB	43	9A	91	25	28	B9																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																									

En este algoritmo el primer bloque es igual al ser en los dos solo ceros, como en el segundo bloque ya se encuentra el uno, cambia, y el resto al depender del antecesor cambian también.

4. Cifrad input.bin e input1.bin con AES-128 en modos ECB , CBC y OFB usando una contraseña a elegir. Explicad los diferentes resultados.

Cifrado:

```
juan@juan-X541UAK:~/Documentos/Spsi/PRACTICAS/PRACTICA1$ openssl enc -aes-128-ecb -in input.bin -pass pass:1234 -out outputecb.bin.enc
juan@juan-X541UAK:~/Documentos/Spsi/PRACTICAS/PRACTICA1$ openssl enc -aes-128-ecb -in input1.bin -pass pass:1234 -out output1ecb.bin.e
nc
```

OutputECB:

```
outputecb.bin.enc ✕
00000000 | 53 61 6C 74 65 64 5F 5F 48 45 34 3B D7 E1 E7 DE 80 A7 | Salted__HE4;.....
00000012 | F1 74 2E BD E6 69 76 4E DB F1 C9 79 9D 70 80 A7 F1 74 | .t...ivN...y.p...t
00000024 | 2E BD E6 69 76 4E DB F1 C9 79 9D 70 80 A7 F1 74 2E BD | ...ivN...y.p...t..
00000036 | E6 69 76 4E DB F1 C9 79 9D 70 80 A7 F1 74 2E BD E6 69 | .ivN...y.p...t...i
00000048 | 76 4E DB F1 C9 79 9D 70 80 A7 F1 74 2E BD E6 69 76 4E | vN...y.p...t...ivN
0000005a | DB F1 C9 79 9D 70 80 A7 F1 74 2E BD E6 69 76 4E DB F1 | ...y.p...t...ivN..
0000006c | C9 79 9D 70 80 A7 F1 74 2E BD E6 69 76 4E DB F1 C9 79 | .y.p...t...ivN...y
0000007e | 9D 70 80 A7 F1 74 2E BD E6 69 76 4E DB F1 C9 79 9D 70 | .p...t...ivN...y.p
00000090 | B6 A3 06 00 2A 6C E0 B9 F5 23 07 67 5F EC D5 07 | .....*l...#.g....
```

Output1ECB:

```
output1ecb.bin.enc ✕
00000000 | 53 61 6C 74 65 64 5F 5F 60 CD C5 B7 7F 6B 4E 1E 73 37 | Salted__`....kN.s7
00000012 | 70 DC 37 14 02 F8 CB 34 39 9E B2 6A 51 C7 A5 CB D3 4B | p.7....49..jQ....K
00000024 | B8 38 BB AC A2 1A 27 83 98 04 4D D1 73 37 70 DC 37 14 | .8....'...M.s7p.7.
00000036 | 02 F8 CB 34 39 9E B2 6A 51 C7 73 37 70 DC 37 14 02 F8 | ...49..jQ.s7p.7...
00000048 | CB 34 39 9E B2 6A 51 C7 73 37 70 DC 37 14 02 F8 CB 34 | .49..jQ.s7p.7....4
0000005a | 39 9E B2 6A 51 C7 73 37 70 DC 37 14 02 F8 CB 34 39 9E | 9..jQ.s7p.7....49.
0000006c | B2 6A 51 C7 73 37 70 DC 37 14 02 F8 CB 34 39 9E B2 6A | .jQ.s7p.7....49..j
0000007e | 51 C7 73 37 70 DC 37 14 02 F8 CB 34 39 9E B2 6A 51 C7 | Q.s7p.7....49..jQ.
00000090 | 2D 7A 31 3A C8 4F B6 62 6B 73 FD 2C D0 F1 04 5C | -z1:.O.bks.,...\
```

Aqui es diferente al cifrar con clave y vector inicial elegido por nosotros, openssl se encarga de cada vez que se cifra elegir una clave y un vector inicial diferente por lo tanto cada vez que se cifre tendrá un resultado diferente.

Cifrado:

```
juan@juan-X541UAK:~/Documentos/Spsi/PRACTICAS/PRACTICA1$ openssl enc -aes-128-ofb -in input.bin -pass pass:1234 -out outputofb.bin.enc
juan@juan-X541UAK:~/Documentos/Spsi/PRACTICAS/PRACTICA1$ openssl enc -aes-128-ofb -in input1.bin -pass pass:1234 -out output1ofb.bin.e
nc
```

## OutputOFB:

outputofb.bin.enc ✖		
00000000	53 61 6C 74 65 64 5F 5F 30 BD E5 DF 87 41 8A 66 F3 C6	Salted__0....A.f..
00000012	A2 8E D1 64 E4 8A F1 F7 4D 3B B0 64 37 E9 5E 3B 6C 27	...d....M;.d7.^;l'
00000024	39 EA 48 51 FB 55 CE D7 C9 2A 3B 53 35 CA 65 6E C7 85	9.HQ.U...*;S5.en..
00000036	00 C4 8A E3 6F 38 0A ED DD AC 49 A4 FF 82 4E D1 66 B0	....o8....I...N.f.
00000048	8D 3B 27 4D 06 9B 7D 2A 76 AD B1 5D B3 89 5C B4 2A D8	.;'M...}*v...].\.*.
0000005a	E4 B4 1D 61 93 D5 69 87 23 B8 D8 0F 00 DB 08 C1 81 75	...a..i.#.....u
0000006c	04 9B 26 7C EB 72 18 8E 30 9C 07 C9 BC 08 49 0E 78 3D	..& .r..0.....I.x=
0000007e	1A 89 75 EF 64 2C E2 22 FE FE 0F 48 57 EC 74 C5 B5 22	..u.d,.."....HW.t..."
00000090		

## Output1OFB:

output1ofb.bin.enc ✖		
00000000	53 61 6C 74 65 64 5F 5F 8B ED 0E 91 0E 4E 89 11 03 BA	Salted_____.N....
00000012	C2 A2 00 72 58 A7 D2 85 93 A9 37 48 9D 30 33 78 0C EF	...rX.....7H.03x..
00000024	A3 09 12 DC DE 3B 51 E0 D9 94 46 BF 29 E2 A2 E2 06 C3	.....;Q...F.).....
00000036	81 24 B8 37 47 B0 86 39 E8 E2 86 81 59 48 E5 73 BF EE	.\$..7G...9....YH.s..
00000048	1F 2C CC 99 D7 75 A1 18 48 E7 5D C4 05 A6 A6 29 1A 68	.....u...H.].)...).h
0000005a	B6 8C 99 20 64 F8 BA 16 43 9E FF 4F 4C A6 D8 47 02 1B	... d...C...OL..G..
0000006c	68 0F 12 31 18 4A 2D BD 60 23 DE 72 0C AE B0 19 3B 9A	h..l.J-..`#.r....;.
0000007e	92 FE 19 6D 19 5E 08 00 12 B2 77 A2 EB 58 5A 93 97 33	...m.^.....w...XZ..3
00000090		

Con este algoritmo pasa exactamente igual que con ECB, openssl cada vez que se cifra cambia aleatoriamente el vector inicial y la clave y ambos archivos son diferentes.

## Cifrado:

```
juan@juan-X541UAK:~/Documentos/Spsi/PRACTICAS/PRACTICA1$ openssl enc -aes-128-cbc -in input.bin -pass pass:1234 -out outputcbc.bin.enc
juan@juan-X541UAK:~/Documentos/Spsi/PRACTICAS/PRACTICA1$ openssl enc -aes-128-cbc -in input1.bin -pass pass:1234 -out output1cbc.bin.e
nc
```

## OutputCBC:

outputcbc.bin.enc ✖		
00000000	53 61 6C 74 65 64 5F 5F 9E 7C 76 7E 4F C8 CD A3 E7 34	Salted_____ v~O....4
00000012	58 BC 7F 9F DE 7F 2C DA C3 96 35 B0 E5 97 51 D4 1E 2B	X.....,...5...Q...+
00000024	93 A1 29 0F 11 DF FE AF 96 E0 A8 BF DB 91 A5 40 05 D5	..).....@..
00000036	89 69 71 40 E6 85 B6 D8 91 90 35 C9 9A 2F 6E D1 B2 F8	.iq@.....5.../n...
00000048	B7 B2 E1 8F 9E D0 63 64 8D 4F 65 DB 2D CC 7A D9 5D D7	.....cd.Oe.-.z.].
0000005a	5A B4 A1 4C 57 E8 FC BE 03 F9 30 64 1B 5B FD 35 09 12	Z..LW.....0d.[.5..
0000006c	BB 04 E5 0F 3F 83 33 9A 1A DE B3 FB 05 02 B5 06 23 C7	....?.3.....#.
0000007e	4B 8B 01 29 65 7F EA 02 09 A1 16 9F 67 F8 EF 47 86 FF	K..)e.....g...G..
00000090	74 98 88 D2 D2 02 16 E0 41 2C 06 26 FB 29 06 0B	t.....A,.&.)..



## Output1CBC:

```
output1cbc.bin.enc ✖
00000000 53 61 6C 74 65 64 5F 5F CB D6 D5 0E 0E 14 65 A7 65 02 |Salted____.e.e.
00000012 3F D7 F7 3A 0C A3 C7 97 25 C0 E6 9D 74 CE B2 7C CB B2 |?.....%...t...|..
00000024 07 A8 4B 29 B6 DC BD 80 77 A2 9D 7C 6C E4 BB 8A 1E 01 |..K)....w...|l....
00000036 69 06 2E E0 47 34 71 FD B2 CA E8 C3 DA DB AE D9 EF BF |i...G4q.....
00000048 25 A0 F7 7D C3 3B B6 58 40 27 8C C0 63 C2 9E B7 AB 8E |%...}.X@'...c.....
0000005a 44 71 9B 7A 59 BA EE 5C 1B B5 3B A0 14 27 D0 44 73 E8 |Dq.zY...\...'.Ds.
0000006c 03 C0 8A 64 61 A9 28 59 04 AD 53 C6 FB 9B C0 BC EC 33 |...da.(Y..S.....3
0000007e 60 25 7C BE 2B 35 94 46 83 51 B7 C5 BE 63 78 6E 36 28 |`%|. +5.F.Q...cxn6(
00000090 ED 51 2C 4F 31 0E 8D 56 96 E5 00 1E 1A 10 56 41 |.Q,O1..V.....VA
```

Y en este pasa igual que en los dos anteriores.

## 5. Repetid el punto anterior con la opción -nosalt.

Al añadir la opción -nosalt al cifrar los ficheros lo que conseguimos es que cada vez que cifremos se utilicen las mismas claves y los mismo vectores de inicialización. Por lo tanto conseguimos que se acabe la aleatoriedad. Abajo se muestran todos los ejemplos:

## Cifrado:

```
juan@juan-X541UAK:~/Documentos/Spsi/PRACTICAS/PRACTICA1$ openssl enc -aes-128-ecb -in input.bin -pass pass:1234 -out outputecbsalt.bin
enc -nosalt
juan@juan-X541UAK:~/Documentos/Spsi/PRACTICAS/PRACTICA1$ openssl enc -aes-128-ecb -in input1.bin -pass pass:1234 -out output1ecbsalt.b
in.enc -nosalt
```

## OutputECB:

```
outputecbsalt.bin.enc ✖
00000000 BE F7 69 70 54 AE 42 F1 B7 98 67 B2 CB F5 9F 50 BE F7 |..ipT.B...g....P..
00000012 69 70 54 AE 42 F1 B7 98 67 B2 CB F5 9F 50 BE F7 69 70 |ipT.B...g....P..ip
00000024 54 AE 42 F1 B7 98 67 B2 CB F5 9F 50 BE F7 69 70 54 AE |T.B...g....P..ipT.
00000036 42 F1 B7 98 67 B2 CB F5 9F 50 BE F7 69 70 54 AE 42 F1 |B...g....P..ipT.B.
00000048 B7 98 67 B2 CB F5 9F 50 BE F7 69 70 54 AE 42 F1 B7 98 |..g....P..ipT.B...
0000005a 67 B2 CB F5 9F 50 BE F7 69 70 54 AE 42 F1 B7 98 67 B2 |g....P..ipT.B...g.
0000006c CB F5 9F 50 BE F7 69 70 54 AE 42 F1 B7 98 67 B2 CB F5 |...P..ipT.B...g...
0000007e 9F 50 DF 3F 8F 28 2C 5D B9 C3 E3 74 3A 56 7A 70 AC 0C |.P.?.(,)...t:Vzp..
00000090
```

## Output1ECB:

```
output1ecbsalt.bin.enc ✖
00000000 BE F7 69 70 54 AE 42 F1 B7 98 67 B2 CB F5 9F 50 24 26 |..ipT.B...g....P$&
00000012 55 DE A9 A6 17 6D A1 1B AD 59 E8 FC 9B E3 BE F7 69 70 |U....m...Y.....ip
00000024 54 AE 42 F1 B7 98 67 B2 CB F5 9F 50 BE F7 69 70 54 AE |T.B...g....P..ipT.
00000036 42 F1 B7 98 67 B2 CB F5 9F 50 BE F7 69 70 54 AE 42 F1 |B...g....P..ipT.B.
00000048 B7 98 67 B2 CB F5 9F 50 BE F7 69 70 54 AE 42 F1 B7 98 |..g....P..ipT.B...
0000005a 67 B2 CB F5 9F 50 BE F7 69 70 54 AE 42 F1 B7 98 67 B2 |g....P..ipT.B...g.
0000006c CB F5 9F 50 BE F7 69 70 54 AE 42 F1 B7 98 67 B2 CB F5 |...P..ipT.B...g...
0000007e 9F 50 DF 3F 8F 28 2C 5D B9 C3 E3 74 3A 56 7A 70 AC 0C |.P.?.(,)...t:Vzp..
00000090
```

## Cifrado:

```
juan@juan-X541UAK:~/Documentos/Spsi/PRACTICAS/PRACTICA1$ openssl enc -aes-128-ofb -in input.bin -pass pass:1234 -out outputofbsalt.bin
enc -nosalt
juan@juan-X541UAK:~/Documentos/Spsi/PRACTICAS/PRACTICA1$ openssl enc -aes-128-ofb -in input1.bin -pass pass:1234 -out output1ofbsalt.b
in.enc -nosalt
```

## OutputOFB:

outputofbsalt.bin.enc ✖	
00000000	18 97 E3 66 42 3A 58 28 4F D2 86 D8 8E FE 1C 9A D7 B0
00000012	FF 82 49 6A B4 BA DB D4 55 38 01 6C 06 CE 6C 5E F4 72
00000024	B2 11 DE B2 5D 00 76 F2 9B 35 4F 4B B3 D3 D7 E5 78 B1
00000036	A5 68 B4 CE 67 6E 5B AE 99 7E 05 3F D5 25 72 AF A3 D4
00000048	08 7C FC EB 1C AF D2 B7 90 70 75 4B EE 9D 19 87 3B 0D
0000005a	2B 51 CC 87 F4 23 EA DA A4 79 66 31 D9 D6 07 3E 8C 94
0000006c	7E C6 F6 85 FB EA A0 57 60 14 E1 0D DD 06 71 E5 90 92
0000007e	1F A3

...fB:X(O.....  
..Ij....U8.l..l^.r  
....].v..5OK....x.  
.h..gn[...~.?.%r...  
.|.....puK....;. .  
+Q...#...yfl...>..  
~.....W`.....q...  
..

## Output1OFB:

output1ofbsalt.bin.enc ✖	
00000000	18 97 E3 66 42 3A 58 28 4F D2 86 D8 8E FE 1C 9A D7 A0
00000012	FF 82 49 6A B4 BA DB D4 55 38 01 6C 06 CE 6C 5E F4 72
00000024	B2 11 DE B2 5D 00 76 F2 9B 35 4F 4B B3 D3 D7 E5 78 B1
00000036	A5 68 B4 CE 67 6E 5B AE 99 7E 05 3F D5 25 72 AF A3 D4
00000048	08 7C FC EB 1C AF D2 B7 90 70 75 4B EE 9D 19 87 3B 0D
0000005a	2B 51 CC 87 F4 23 EA DA A4 79 66 31 D9 D6 07 3E 8C 94
0000006c	7E C6 F6 85 FB EA A0 57 60 14 E1 0D DD 06 71 E5 90 92
0000007e	1F A3

...fB:X(O.....  
..Ij....U8.l..l^.r  
....].v..5OK....x.  
.h..gn[...~.?.%r...  
.|.....puK....;. .  
+Q...#...yfl...>..  
~.....W`.....q...  
..

## Cifrado:

```
juan@juan-X541UAK:~/Documentos/Spsi/PRACTICAS/PRACTICA1$ openssl enc -aes-128-cbc -in input.bin -pass pass:1234 -out outputcbcsalt.bin
enc -nosalt
juan@juan-X541UAK:~/Documentos/Spsi/PRACTICAS/PRACTICA1$ openssl enc -aes-128-cbc -in input1.bin -pass pass:1234 -out output1cbcsalt.b
in.enc -nosalt
```

## OutputCBC:

outputcbcsalt.bin.enc ✖	
00000000	18 97 E3 66 42 3A 58 28 4F D2 86 D8 8E FE 1C 9A D7 B0
00000012	FF 82 49 6A B4 BA DB D4 55 38 01 6C 06 CE 6C 5E F4 72
00000024	B2 11 DE B2 5D 00 76 F2 9B 35 4F 4B B3 D3 D7 E5 78 B1
00000036	A5 68 B4 CE 67 6E 5B AE 99 7E 05 3F D5 25 72 AF A3 D4
00000048	08 7C FC EB 1C AF D2 B7 90 70 75 4B EE 9D 19 87 3B 0D
0000005a	2B 51 CC 87 F4 23 EA DA A4 79 66 31 D9 D6 07 3E 8C 94
0000006c	7E C6 F6 85 FB EA A0 57 60 14 E1 0D DD 06 71 E5 90 92
0000007e	1F A3 A0 E0 F3 07 D2 37 FB 76 E2 BA CA 5C 11 00 D3 B0
00000090	

...fB:X(O.....  
..Ij....U8.l..l^.r  
....].v..5OK....x.  
.h..gn[...~.?.%r...  
.|.....puK....;. .  
+Q...#...yfl...>..  
~.....W`.....q...  
.....7.v...\....



## Output1CBC:

```
output1cbcsalt.bin.enc ✕
00000000 | 18 97 E3 66 42 3A 58 28 4F D2 86 D8 8E FE 1C 9A 9A 4E | ...fB:X(O.....N
00000012 | DC F4 D5 28 EA D3 E2 6B AD 9D 42 38 62 84 91 A0 93 38 | ... (...k..B8b....8
00000024 | 63 36 23 18 12 19 F3 9D 8A 41 79 F6 51 E1 DB B4 F5 2C | c6#.....Ay.Q....,
00000036 | 7F A8 09 EE 0A 90 7A 68 2D D3 D3 7F E5 BF C9 4D EA 6A | .....zh-.....M.j
00000048 | 6D CA F2 5B 5F 05 C1 72 40 4A 82 47 92 C2 2C 65 8D 61 | m..[_..r@J.G...,e.a
0000005a | 14 25 C0 EE 6B 33 54 90 80 CA 0C E6 C3 05 EF 02 31 A3 | .%...k3T.....1.
0000006c | F6 3E D5 F0 51 00 AB DF 20 98 03 A4 F3 15 01 1C 94 57 | .>...Q... .....W
0000007e | 7A E6 2D F2 3B 83 68 74 93 E7 3D 7F DF B4 77 73 43 31 | z.-.;.ht..=...wsC1
00000090
```

6. Cifrad input.bin con AES-192 en modo OFB ,clave y vector de inicialización a elegir (no contraseña). Supongamos que la salida es output.bin .

## Cifrado:

```
juan@juan-X541UAK:~/Documentos/Spsi/PRACTICAS/PRACTICA1$ openssl enc -aes-192-ofb -in input.bin -out outputofb6.bin -K 20e364d2ccf7ce13f5bc49a -iv acdf76ca965fd65b976df7a
```

## Output:

```
outputofb6.bin ✕
00000000 | DA 51 82 F2 19 65 69 49 52 09 4C A5 4D B0 44 9D 34 01 | .Q...eiIR.L.M.D.4.
00000012 | 08 6C 9A F8 EB BF F7 06 F9 1E 56 DE 7B 39 C9 96 48 E0 | .l.....V.{9..H.
00000024 | 76 CA 9D F7 19 35 20 F3 F1 AA A8 82 BC 48 CA 8B B4 04 | v....5 .....H....
00000036 | A8 B3 CD EE 97 8B 01 B5 5B 61 15 93 BA FB 04 C0 5B F7 | .....[a.....[.
00000048 | C1 B8 C6 36 CE 6D 47 4E CC CD F5 AB 86 0F F1 A8 F3 E9 | ...6.mGN.....
0000005a | 77 2A 35 96 F4 35 33 CD E3 56 A8 2F 4A B9 6C 06 2E 8A | w*5...53..V./J.l...
0000006c | 5B A4 89 61 42 7F 45 4F 1A E7 CF 74 DB C1 D4 6E 75 CE | [...aB.EO...t...nu.
0000007e | 1E B5 ..
```

7. Descifrad output.bin utilizando la misma clave y vector de inicialización que en 6.

Para descifrar lo único que tenemos que añadir es el parámetro -d:

## Descifrado:

```
juan@juan-X541UAK:~/Documentos/Spsi/PRACTICAS/PRACTICA1$ openssl aes-192-ofb -d -in outputofb6.bin -out descifrado -K 20e364d2ccf7ce13f5bc49a -iv acdf76ca965fd65b976df7a
```

## Descifrado (Input.bin):

```
descifrado ✕
00000000 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00000012 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00000024 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00000036 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00000048 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0000005a 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0000006c 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0000007e 00 00 ..
```

8. Vuelve a cifrar output.bin con AES-192 en modo OFB , clave y vector de inicialización del punto 6. Compara el resultado obtenido con el punto 7, explicando el resultado.

## Cifrado:

```
juan@juan-X541UAK:~/Documentos/Spsi/PRACTICAS/PRACTICA1$ openssl enc -aes-192-ofb -in outputofb6.bin -out outputofb8.bin -K 20e364d2ccf7ce13f5bc49a -iv acdf76ca965fd65b976df7a
```

## Output:

```
outputofb8.bin ✕
00000000 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00000012 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00000024 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00000036 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00000048 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0000005a 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0000006c 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0000007e 00 00 ..
```

Al realizarlo y compararlo, podemos ver que el resultado es el archivo inicial. Esto sucede al utilizar el operador XOR, se cifra carácter a carácter con la misma clave, y por tanto si el resultado se vuelve a cifrar con la misma clave, conseguimos volver al archivo inicial.

9. Repite los puntos 6 al 8 pero empleando contraseña en lugar de clave y vector de inicialización.

## Paso 6 (Cifrado):

```
juan@juan-X541UAK:~/Documentos/Spsi/PRACTICAS/PRACTICA1$ openssl enc -aes-192-ofb -in input.bin -out outputofb9.bin -pass pass:1234
```

Output:

```
outputofb9.bin x
00000000 53 61 6C 74 65 64 5F 5F 46 6B 4B F7 FB 21 65 C7 3F 84 | Salted__FkK..!e.?.
00000012 F7 0A 5A 62 99 F8 9D 06 5D 87 28 F4 F6 54 2A F4 C0 43 | ..Zb....].(..T*..C
00000024 0B 7B 88 6B FE 52 17 74 6A 72 8E CE CD 9E A9 10 1C 4D | {.k.R.tjr.....M
00000036 CF 9E 7E 4A 45 73 67 5A 22 80 71 BD 1A D7 1E 4C 42 CA | ..~JEsGZ".q....LB.
00000048 D8 2A 96 5C 86 2E F7 7D 9F A9 A8 3A 2C 8E A7 34 EC A8 | .*.\....}....:,..4..
0000005a 89 08 C4 8E 26 24 63 A4 46 11 1F 17 61 F1 8C 87 98 7B | ....&$c.F...a....{
0000006c 55 AB DE EE 60 F6 B4 E2 E8 1E 25 74 DC 57 D0 19 06 00 | U...`.....%t.W....
0000007e 1E 23 BB 7E 9E A8 4C DA AA F9 36 BE 6B 98 8C B1 D3 8C | .#.~..L...6.k.....
00000090
```

Paso 7 (Descifrado):

```
juan@juan-X541UAK:~/Documentos/Spsi/PRACTICAS/PRACTICA1$ openssl aes-192-ofb -d -in outputofb9.bin -out descifrado9
enter aes-192-ofb decryption password:
```

Descifrado:

```
descifrado9 x
00000000 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
00000012 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
00000024 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
00000036 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
00000048 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
0000005a 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
0000006c 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
0000007e 00 00 | ..
```

Paso 8:

```
juan@juan-X541UAK:~/Documentos/Spsi/PRACTICAS/PRACTICA1$ openssl enc -aes-192-ofb -in outputofb9.bin -out outputofb98.bin -pass pass:1234
```

```
outputofb98.bin x
00000000 53 61 6C 74 65 64 5F 5F CB BB 16 40 C8 CC 4E 5D EA 3C | Salted____@...N].<
00000012 8A 3B 62 CF 51 FF 73 4B FA 65 0B CB 83 94 7E 69 05 C5 | ;b.Q.sK.e....~i..
00000024 C5 3D EA E7 78 BA 63 2A 9C 70 42 6C 9C 5F C7 DE 01 E6 | .=..x.c*.pBl._....
00000036 EA BC 7E D4 80 88 46 07 C5 BA A3 29 2B AB AB DC 62 EF | ..~...F....)+...b.
00000048 15 4C 3E 46 3D 2D 5B 29 4C 5C 9F EF C1 A2 EB 9E BE AC | .L>F=-()L\.....
0000005a 57 AB D6 0F C2 6D 4F A6 7B 0C 76 2D D9 14 00 36 45 74 | W....mO.{.v-...6Et
0000006c 04 30 9D 03 B0 77 80 24 A3 15 66 37 C8 0C 00 2F C4 90 | .0...w.$..f7.../..
0000007e A9 4B 69 6E E2 BD 84 8A 46 F9 AC 47 05 3A 96 67 66 2D | .Kin....F..G.:.gf-
00000090 F6 0B BC E3 A2 C1 81 C7 EA 2C 0D B2 AC 7D 85 A6 | .....,...}..
```

Aquí no pasa como en el ejercicio anterior ya que no se cifra con la misma clave, cada vez que se cifra se utiliza una clave aleatoria y por tanto al volver a cifrar el archivo será diferente.

10. Presentad la descripción de otro algoritmo de cifrado simétrico que aparezca en vuestra implementación de OpenSSL .

En este caso el algoritmo de encriptación que voy a presentar es el RC2.

```
-rc2          -rc2-40-cbc      -rc2-64-cbc
-rc2-cbc      -rc2-cfb         -rc2-ecb
-rc2-ofb      -rc4             -rc4-40
```

Es un cifrado de bloque de clave simétrica diseñado por Ron Rivest en 1987, este algoritmo pertenece a a categoría de cifradores de bloque que emplean una función de feistel, el método que sigue es el mismo que en los demás algoritmos, la división en bloques, la aplicación de s-cajas y la función XOR.

La diferencia de este algoritmo con los otros es que aunque usa una clave con bloque de 64 bits, realmente la longitud de la clave puede ser variable, es decir no obliga a que sean exactamente claves de 64 bits, lo cual dificulta su decodificación, aunque cabe mencionar que este algoritmo ya no es empleado debido a que ya ha sido vulnerado y fue revelado en Internet de forma anónima.