

## Práctica 1. Criptosistemas simétricos

---

Patricia Maldonado Mancilla

# Índice

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 podéis dar el nombre que os convenga.	6
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	6
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	7
3.1. Cifrado AES-256 modo ECB . . . . .	7
3.2. Cifrado AES-256 modo CBC . . . . .	8
3.3. Cifrado AES-256 modo OFB . . . . .	9
3.4. Cifrado AES-256 modo ECB, CBC y OFB input . . . . .	11
3.5. Cifrado AES-256 modo ECB, CBC y OFB input 1 . . . . .	12
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.	13
4.1. Cifrado AES-128 modo ECB . . . . .	13
4.2. Cifrado AES-128 modo CBC . . . . .	14
4.3. Cifrado AES-128 modo OFB . . . . .	14
5. Repetid el punto anterior con la opción -nosalt	15
5.1. Cifrado AES-128 modo ECB opción -nosalt . . . . .	15
5.2. Cifrado AES-128 modo CBC opción -nosalt . . . . .	16
5.3. Cifrado AES-128 modo OFB opción -nosalt . . . . .	17
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	18
7. Descrifrad output.bin utilizando a misma clave y vector de inicialización que en 6.	19
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	20
9. Repite los puntos 6 al 8 pero empleando contraseña en lugar de clave y vector de inicialización.	20

<b>10. Presentad la descripción de otro algoritmo de cifrado simétrico que aparezca en vuestra implementación de OpenSSL.</b>	<b>22</b>
<b>11. Repetid los puntos de 3 a 5 con el cifrado presentado en el punto 10.</b>	<b>23</b>
11.1. Cifrado input.bin e input1.bin con Camellia-128 en modos ECB, CBC y OFB usando una clave. . . . .	23
11.1.1. Cifrado Camellia-128 modo ECB . . . . .	23
11.1.2. Cifrado Camellia-128 modo CBC . . . . .	24
11.1.3. Cifrado Camellia-128 modo OFB . . . . .	25
11.2. Cifrado input.bin e input1.bin con Camellia-128 en modos ECB, CBC y OFB usando contraseña. . . . .	26
11.2.1. Cifrado Camellia-128 modo ECB con contraseña . . . . .	26
11.2.2. Cifrado Camellia-128 modo CBC con contraseña . . . . .	27
11.2.3. Cifrado Camellia-128 modo OFB con contraseña . . . . .	28
11.3. Repetición del punto anterior con la opción -nosalt . . . . .	29
11.3.1. Cifrado Camellia-128 modo ECB con contraseña opción -nosalt . .	29
11.3.2. Cifrado Camellia-128 modo CBC con contraseña opción -nosalt . .	30
11.3.3. Cifrado Camellia-128 modo OFB con contraseña opción -nosalt . .	31

## Índice de figuras

1.1. Contenido del archivo input.bin . . . . .	6
2.1. Contenido del archivo input1.bin . . . . .	6
3.1. Comando para cifrar con AES-256 modo ECB . . . . .	7
3.2. input AES-256 modo ECB . . . . .	8
3.3. input1 AES-256 modo ECB . . . . .	8
3.4. Comando para cifrar con AES-256 modo CBC . . . . .	8
3.5. input AES-256 modo CBC . . . . .	9
3.6. input1 AES-256 modo CBC . . . . .	9
3.7. Comando para cifrar con AES-256 modo OFB . . . . .	9
3.8. input AES-256 modo OFB . . . . .	10
3.9. input1 AES-256 modo OFB . . . . .	10
3.10. input AES-256 modo ECB . . . . .	11
3.11. input AES-256 modo CBC . . . . .	11
3.12. input AES-256 modo OFB . . . . .	11
3.13. input1 AES-256 modo ECB . . . . .	12
3.14. input1 AES-256 modo CBC . . . . .	12
3.15. input1 AES-256 modo OFB . . . . .	12
4.1. Comando para cifrar con AES-128 modo ECB . . . . .	13
4.2. input AES-128 modo ECB . . . . .	13
4.3. input1 AES-128 modo ECB . . . . .	13
4.4. Comando para cifrar con AES-128 modo CBC . . . . .	14
4.5. input AES-128 modo CBC . . . . .	14

4.6.	input1 AES-128 modo CBC . . . . .	14
4.7.	Comando para cifrar con AES-128 modo OFB . . . . .	14
4.8.	input AES-128 modo OFB . . . . .	15
4.9.	input1 AES-128 modo OFB . . . . .	15
5.1.	Comando para cifrar con AES-128 modo ECB -nosalt . . . . .	15
5.2.	input AES-128 modo ECB -nosalt . . . . .	16
5.3.	input1 AES-128 modo ECB -nosalt . . . . .	16
5.4.	Comando para cifrar con AES-128 modo CBC -nosalt . . . . .	16
5.5.	input AES-128 modo CBC -nosalt . . . . .	17
5.6.	input1 AES-128 modo CBC -nosalt . . . . .	17
5.7.	Comando para cifrar con AES-128 modo OFB -nosalt . . . . .	17
5.8.	input AES-128 modo OFB -nosalt . . . . .	18
5.9.	input1 AES-128 modo OFB -nosalt . . . . .	18
6.1.	Comando para cifrar con AES-192 modo OFB . . . . .	18
6.2.	input AES-192 modo OFB . . . . .	19
7.1.	Comando para descifrar con AES-192 modo OFB . . . . .	19
7.2.	output.bin descifrado AES-192 modo OFB . . . . .	19
8.1.	Comando para cifrar con AES-192 modo OFB . . . . .	20
8.2.	output.bin cifrado AES-192 modo OFB . . . . .	20
9.1.	Comando para cifrar con AES-192 modo OFB con contraseña . . . . .	20
9.2.	input.bin cifrado AES-192 modo OFB con contraseña . . . . .	21
9.3.	Comando para descifrar con AES-192 modo OFB con contraseña . . . . .	21
9.4.	output2.bin descifrado AES-192 modo OFB con contraseña . . . . .	21
9.5.	Comando para cifrar con AES-192 modo OFB con contraseña . . . . .	21
9.6.	output2.bin cifrado AES-192 modo OFB con contraseña . . . . .	22
10.1.	Algoritmos de mi implementación de OpenSSL . . . . .	22
11.1.	Comando Camellia-128 modo ECB . . . . .	23
11.2.	input camellia-128 modo ECB . . . . .	24
11.3.	input1 camellia-128 modo ECB . . . . .	24
11.4.	Comando Camelia-128 modo CBC . . . . .	24
11.5.	input camellia-128 modo CBC . . . . .	25
11.6.	input1 camellia-128 modo CBC . . . . .	25
11.7.	Comando Camelia-128 modo OFB . . . . .	25
11.8.	input camellia-128 modo OFB . . . . .	26
11.9.	input1 camellia-128 modo OFB . . . . .	26
11.10.	Comando Camelia-128 modo ECB con contraseña . . . . .	26
11.11.	input camellia-128 modo ECB con contraseña . . . . .	27
11.12.	input1 camellia-128 modo ECB contraseña . . . . .	27
11.13.	Comando Camelia-128 modo CBC con contraseña . . . . .	27
11.14.	input camellia-128 modo CBC con contraseña . . . . .	28
11.15.	input1 camellia-128 modo CBC contraseña . . . . .	28
11.16.	Comando Camelia-128 modo OFB con contraseña . . . . .	28
11.17.	input camellia-128 modo OFB con contraseña . . . . .	29
11.18.	input1 camellia-128 modo OFB contraseña . . . . .	29

11.19	Comando Camelia-128 modo ECB con contraseña . . . . .	29
11.20	input camellia-128 modo ECB con contraseña opción -nosalt . . . . .	30
11.21	input1 camellia-128 modo ECB contraseña opción -nosalt . . . . .	30
11.22	Comando Camelia-128 modo OFB con contraseña . . . . .	30
11.23	input camellia-128 modo CBC con contraseña opción -nosalt . . . . .	31
11.24	input1 camellia-128 modo CBC contraseña opción -nosalt . . . . .	31
11.25	Comando Camelia-128 modo OFB con contraseña . . . . .	31
11.26	input camellia-128 modo OFB con contraseña opción -nosalt . . . . .	32
11.27	input1 camellia-128 modo OFB contraseña opción -nosalt . . . . .	32

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 podéis dar el nombre que os convenga.

Para empezar creo el archivo binario `input.bin`, para ello he usado la herramienta `ghex` y lo he rellenado con ceros hasta llegar hasta 1024 bits.

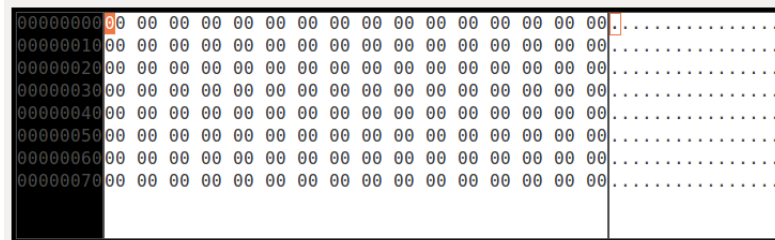


Figura 1.1: Contenido del archivo `input.bin`

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`

En este caso he creado el archivo y lo he rellenado con ceros menos un bit con valor 1 entre los bits 130 y 150. Podemos ver que el bit 1 se encuentra en el segundo bloque.

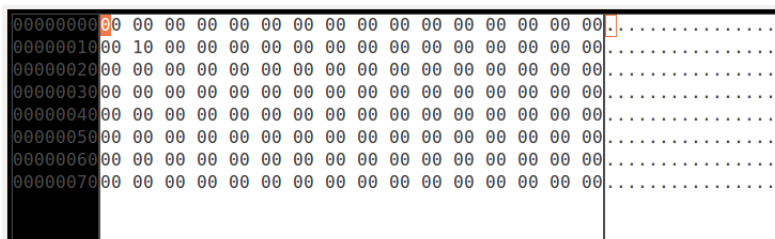


Figura 2.1: Contenido del archivo `input1.bin`

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

**AES (Advanced Encryption Standard)** es un cifrado simétrico por bloques, cifra y descifra los datos en bloques de 128 bits cada uno. Para ello, utiliza una clave criptográfica, que es un conjunto de protocolos para manipular información. Esta clave puede ser de 128, 192 o 256 bits de tamaño. En este caso vamos a utilizar la clave de 256 bits, la más avanzada del cifrado y consiste en 14 rondas de sustitución, transposición y mezcla para un nivel de seguridad excepcionalmente alto.

### 3.1. Cifrado AES-256 modo ECB

Para cifrar con AES-256 en modo ECB usamos el siguiente comando.

[illegible]

Figura 3.1: Comando para cifrar con AES-256 modo ECB

Los argumentos significan lo siguiente:

-enc -aes-256-ecb: tipo de cifrado simétrico. AES-256 hace referencia al tamaño de la clave que debe ser de 256 bits.

-in: fichero de entrada.

-out: fichero de salida.

- K: clave para el cifrado.







00000000	33	D3	EE	91	1D	75	C3	DC	A3	AA	CC	E5	FC	32	0C	C2	33	....u.....2..
00000010	FB	6C	B8	62	98	47	58	31	C0	53	EB	79	DB	0B	6E	B8	.l.b.GX1.S.y..n.	
00000020	03	3C	D6	61	86	BA	94	D3	B8	AC	BA	A6	67	0D	39	EA	.<.a.....g.9.	
00000030	0C	49	0C	EE	22	C6	15	CB	88	CE	DB	50	2D	CB	BD	1D	.I.."......P-...	
00000040	0F	78	13	0D	53	64	9D	B2	EE	17	07	26	CC	9D	2E	BC	.x..Sd.....&....	
00000050	93	8D	76	B7	FF	75	C1	FB	48	B4	4A	81	BF	22	A2	25	..v...u..H.J..".%	
00000060	96	7B	EC	34	2C	61	32	55	9B	E8	07	61	EB	A0	6B	F1	.{.4,a2U...a..k.	
00000070	7A	FF	31	8C	B5	26	28	BB	A2	35	71	A8	4D	59	E3	20	z.1..&(..5q.MY.	

Figura 3.8: input AES-256 modo OFB

00000000	33	D3	EE	91	1D	75	C3	DC	A3	AA	CC	E5	FC	32	0C	C2	33	....u.....2..
00000010	FB	7C	B8	62	98	47	58	31	C0	53	EB	79	DB	0B	6E	B8	. .b.GX1.S.y..n.	
00000020	03	3C	D6	61	86	BA	94	D3	B8	AC	BA	A6	67	0D	39	EA	.<.a.....g.9.	
00000030	0C	49	0C	EE	22	C6	15	CB	88	CE	DB	50	2D	CB	BD	1D	.I.."......P-...	
00000040	0F	78	13	0D	53	64	9D	B2	EE	17	07	26	CC	9D	2E	BC	.x..Sd.....&....	
00000050	93	8D	76	B7	FF	75	C1	FB	48	B4	4A	81	BF	22	A2	25	..v...u..H.J..".%	
00000060	96	7B	EC	34	2C	61	32	55	9B	E8	07	61	EB	A0	6B	F1	.{.4,a2U...a..k.	
00000070	7A	FF	31	8C	B5	26	28	BB	A2	35	71	A8	4D	59	E3	20	z.1..&(..5q.MY.	

Figura 3.9: input1 AES-256 modo OFB

El modo OFB hace que el cifrado en bloque opere como una unidad de flujo de cifrado: se generan bloques de flujo de claves, que son operados con XOR y el texto en claro para obtener el texto cifrado. En OFB al intercambiar un bit en el texto cifrado produce texto cifrado con un bit intercambiado en el texto plano en la misma ubicación. Podemos ver como cambia justo en el lugar donde hemos puesto el bit a 1 en el archivo input1. Además este modo no tiene padding como en los modos anteriores.

### 3.4. Cifrado AES-256 modo ECB, CBC y OFB input

00000000	7A E6 39 B9 ED A3 61 AE C8 DA 69 B6 16 B5 C2 D7	z.9...a...i....
00000010	7A E6 39 B9 ED A3 61 AE C8 DA 69 B6 16 B5 C2 D7	z.9...a...i....
00000020	7A E6 39 B9 ED A3 61 AE C8 DA 69 B6 16 B5 C2 D7	z.9...a...i....
00000030	7A E6 39 B9 ED A3 61 AE C8 DA 69 B6 16 B5 C2 D7	z.9...a...i....
00000040	7A E6 39 B9 ED A3 61 AE C8 DA 69 B6 16 B5 C2 D7	z.9...a...i....
00000050	7A E6 39 B9 ED A3 61 AE C8 DA 69 B6 16 B5 C2 D7	z.9...a...i....
00000060	7A E6 39 B9 ED A3 61 AE C8 DA 69 B6 16 B5 C2 D7	z.9...a...i....
00000070	7A E6 39 B9 ED A3 61 AE C8 DA 69 B6 16 B5 C2 D7	z.9...a...i....
00000080	EB 03 4A C1 AA 10 51 28 0A 5C 99 21 17 CF DA 01	..J...Q(.\!....

Figura 3.10: input AES-256 modo ECB

00000000	3B D3 EE 91 1D 75 C3 DC A3 AA CC E5 FC 32 0C C2	3....u.....2..
00000010	FB 6C B8 62 98 47 58 31 C0 53 EB 79 DB 0B 6E B8	.l.b.GX1.S.y..n.
00000020	03 3C D6 61 86 BA 94 D3 B8 AC BA A6 67 0D 39 EA	.<.a.....g.9.
00000030	0C 49 0C EE 22 C6 15 CB 88 CE DB 50 2D CB BD 1D	.I.."......P-...
00000040	0F 78 13 0D 53 64 9D B2 EE 17 07 26 CC 9D 2E BC	.x..Sd.....&....
00000050	93 8D 76 B7 FF 75 C1 FB 48 B4 4A 81 BF 22 A2 25	..v...u..H.J..".%
00000060	96 7B EC 34 2C 61 32 55 9B E8 07 61 EB A0 6B F1	.{.4,a2U...a..k.
00000070	7A FF 31 8C B5 26 28 BB A2 35 71 A8 4D 59 E3 20	z.1..&(.5q.MY.
00000080	B6 75 50 0A 5F 9A F4 2D 49 56 70 E3 E6 06 E4 95	.uP._...-IVp.....

Figura 3.11: input AES-256 modo CBC

00000000	3B D3 EE 91 1D 75 C3 DC A3 AA CC E5 FC 32 0C C2	3....u.....2..
00000010	FB 6C B8 62 98 47 58 31 C0 53 EB 79 DB 0B 6E B8	.l.b.GX1.S.y..n.
00000020	03 3C D6 61 86 BA 94 D3 B8 AC BA A6 67 0D 39 EA	.<.a.....g.9.
00000030	0C 49 0C EE 22 C6 15 CB 88 CE DB 50 2D CB BD 1D	.I.."......P-...
00000040	0F 78 13 0D 53 64 9D B2 EE 17 07 26 CC 9D 2E BC	.x..Sd.....&....
00000050	93 8D 76 B7 FF 75 C1 FB 48 B4 4A 81 BF 22 A2 25	..v...u..H.J..".%
00000060	96 7B EC 34 2C 61 32 55 9B E8 07 61 EB A0 6B F1	.{.4,a2U...a..k.
00000070	7A FF 31 8C B5 26 28 BB A2 35 71 A8 4D 59 E3 20	z.1..&(.5q.MY.

Figura 3.12: input AES-256 modo OFB

Para la primera imagen con un cifrado AES-256 modo ECB, AES se divide en bloques de 128 bits. Podemos ver como cada bloque se repite exactamente igual, ya que ECB cifra por bloques independientes y todos los bloques contenian 0 por lo que seran iguales menos el último bloque, que es lo que se conoce como padding. Esta técnica completa el tamaño para que coincida con un múltiplo de tamaño del bloque. Este bloque podría eliminarse con la opción -nopad, siempre y cuando estemos seguros de que nuestro archivo tiene un múltiplo exacto del tamaño de bloque.

Para CBC y OFB vemos como la salida del cifrado es igual, la única diferencia, es que en el modo CBC hay padding, en cambio en el modo OFB no necesita padding ya que

utiliza un algoritmo de cifrado para generar lo que llamamos un flujo de clave, que se mezclará con el texto plano para ir produciendo texto cifrado. Esto convierte el cifrado por bloques en un cifrado de flujo, por lo que se elimina la necesidad de padding. Para dos entradas iguales en diferentes modos deberían de salir resultados diferentes, pero esto no ocurre en este caso, ya que al estar contenido de ceros en ambos casos no produce ningún cambio, al no cambiar ningún bit.

### 3.5. Cifrado AES-256 modo ECB, CBC y OFB input 1

00000000	7A E6 39 B9 ED A3 61 AE C8 DA 69 B6 16 B5 C2 D7	z.9...a...i....
00000010	0A 91 D3 C2 19 6A 2F 73 7A 35 19 C3 93 72 31 4F	.....j/sz5...r10
00000020	7A E6 39 B9 ED A3 61 AE C8 DA 69 B6 16 B5 C2 D7	z.9...a...i....
00000030	7A E6 39 B9 ED A3 61 AE C8 DA 69 B6 16 B5 C2 D7	z.9...a...i....
00000040	7A E6 39 B9 ED A3 61 AE C8 DA 69 B6 16 B5 C2 D7	z.9...a...i....
00000050	7A E6 39 B9 ED A3 61 AE C8 DA 69 B6 16 B5 C2 D7	z.9...a...i....
00000060	7A E6 39 B9 ED A3 61 AE C8 DA 69 B6 16 B5 C2 D7	z.9...a...i....
00000070	7A E6 39 B9 ED A3 61 AE C8 DA 69 B6 16 B5 C2 D7	z.9...a...i....
00000080	EB 03 4A C1 AA 10 51 28 0A 5C 99 21 17 CF DA 01	..J...Q(.\.!....

Figura 3.13: input1 AES-256 modo ECB

00000000	33 D3 EE 91 1D 75 C3 DC A3 AA CC E5 FC 32 0C C2	3....u.....2..
00000010	42 F3 D4 9F E7 C1 02 D7 E8 E7 09 83 82 B3 CC EE	B.....
00000020	C2 4C D5 84 E6 5A D5 A6 E1 A8 C9 9B CF E3 30 6F	.L...Z.....0o
00000030	D4 7F D8 CB 7C DB F5 A4 D0 3B 90 93 48 98 62 EE	.... ...;..H.b.
00000040	67 C6 DE 74 68 35 B8 88 2E 2C BF BF D7 F0 59 14	g..th5....Y.
00000050	CC 99 5B 06 1E 5F 9D 27 EA 63 4D 98 CF 99 0B 74	..[...'.cM....t
00000060	8F 83 6A B8 AA 99 9C 98 BC 55 91 FB A8 22 E5 10	..j.....U..."..
00000070	7C 99 8C 07 45 52 EE C0 FE DA E5 A2 F6 7C 39 7E	...ER..... 9~
00000080	D4 6E C3 E6 5F CB D1 1F 36 C7 56 EC 0D 9B D6 84	.n..._...6.V....

Figura 3.14: input1 AES-256 modo CBC

00000000	33 D3 EE 91 1D 75 C3 DC A3 AA CC E5 FC 32 0C C2	3....u.....2..
00000010	FB 7C B8 62 98 47 58 31 C0 53 EB 79 DB 0B 6E B8	. .b.GX1.S.y.n.
00000020	03 3C D6 61 86 BA 94 D3 B8 AC BA A6 67 0D 39 EA	<.a.....g.9.
00000030	0C 49 0C EE 22 C6 15 CB 88 CE DB 50 2D CB BD 1D	.I...".....P-...
00000040	0F 78 13 0D 53 64 9D B2 EE 17 07 26 CC 9D 2E BC	.x..Sd.....&....
00000050	93 8D 76 B7 FF 75 C1 FB 48 B4 4A 81 BF 22 A2 25	..v...u..H.J..."%
00000060	96 7B EC 34 2C 61 32 55 9B E8 07 61 EB A0 6B F1	{.4,a2U...a.k.
00000070	7A FF 31 8C B5 26 28 BB A2 35 71 A8 4D 59 E3 20	z.1..&(..5q.MY.

Figura 3.15: input1 AES-256 modo OFB

Para la primera imagen con un cifrado AES-256 modo ECB, AES se divide en bloques de 128 bits. Podemos ver como cada bloque se repite exactamente igual menos el segundo

bloque, ya que ECB cifra por bloques y todos los bloques contenian 0 menos el bit a 1 que hemos modificado en el segundo bloque. Además al igual que pasa en input1 con este cifrado, tambien contiene padding en el último bloque. En los siguientes modos CBC y OFB cambian a partir del segundo bloque en el que hemos cambiado el bit a 1, en este caso al hacer este cambio si tenemos salidas diferentes por la forma de cifrar de cada modo y su encadenamiento con los bloques anteriores.

#### 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.

##### 4.1. Cifrado AES-128 modo ECB

Para cifrar con AES-128 en modo ECB con contraseña usamos el siguiente comando.

```
patri@patri:~/Escritorio$ openssl enc -aes-128-ecb -in input1.bin -pass pass:123456 -out input1ecb_aes128.enc
```

Figura 4.1: Comando para cifrar con AES-128 modo ECB

En este caso utilizamos el argumento -pass pass:(contraseñadeseada) para cifrar nuestro archivo con contraseña.

```
00000000 53 61 6C 74 65 64 5F 5F B7 3B 7D 0B 64 6F A0 94 Salted __.};.do..
00000010 AC 38 63 66 27 49 08 FB 19 F7 2A CD 88 E2 DF D4 .8cf'I....*.....
00000020 AC 38 63 66 27 49 08 FB 19 F7 2A CD 88 E2 DF D4 .8cf'I....*.....
00000030 AC 38 63 66 27 49 08 FB 19 F7 2A CD 88 E2 DF D4 .8cf'I....*.....
00000040 AC 38 63 66 27 49 08 FB 19 F7 2A CD 88 E2 DF D4 .8cf'I....*.....
00000050 AC 38 63 66 27 49 08 FB 19 F7 2A CD 88 E2 DF D4 .8cf'I....*.....
00000060 AC 38 63 66 27 49 08 FB 19 F7 2A CD 88 E2 DF D4 .8cf'I....*.....
00000070 AC 38 63 66 27 49 08 FB 19 F7 2A CD 88 E2 DF D4 .8cf'I....*.....
00000080 AC 38 63 66 27 49 08 FB 19 F7 2A CD 88 E2 DF D4 .8cf'I....*.....
00000090 4D 4A 33 A9 1E BF 0B 8E 07 C5 3A 80 FD 4B D1 78 MJ3.....:..K.x
```

Figura 4.2: input AES-128 modo ECB

```
00000000 53 61 6C 74 65 64 5F 5F A4 25 32 DF BE 37 BF D0 Salted __.%2..7..
00000010 FE 2F 7A A8 1A 70 8D 43 C2 93 CE DC 14 E0 6C 01 ./z..p.C.....l.
00000020 4B 4C 56 2D F4 CB B6 35 22 81 96 8D 1D FD 69 92 KLV-...5".....i.
00000030 FE 2F 7A A8 1A 70 8D 43 C2 93 CE DC 14 E0 6C 01 ./z..p.C.....l.
00000040 FE 2F 7A A8 1A 70 8D 43 C2 93 CE DC 14 E0 6C 01 ./z..p.C.....l.
00000050 FE 2F 7A A8 1A 70 8D 43 C2 93 CE DC 14 E0 6C 01 ./z..p.C.....l.
00000060 FE 2F 7A A8 1A 70 8D 43 C2 93 CE DC 14 E0 6C 01 ./z..p.C.....l.
00000070 FE 2F 7A A8 1A 70 8D 43 C2 93 CE DC 14 E0 6C 01 ./z..p.C.....l.
00000080 FE 2F 7A A8 1A 70 8D 43 C2 93 CE DC 14 E0 6C 01 ./z..p.C.....l.
00000090 D7 EA 78 A2 8F 88 A6 F4 B6 44 3C 3A 16 E7 5B E0 ..x.....D<:..[.
```

Figura 4.3: input1 AES-128 modo ECB

## 4.2. Cifrado AES-128 modo CBC

Para cifrar con AES-128 en modo CBC con contraseña usamos el siguiente comando.

```
patri@patri:~/Escritorio$ openssl enc -aes-128-cbc -in input.bin -pass pass:123456 -out inputcbc_aes128.enc
```

Figura 4.4: Comando para cifrar con AES-128 modo CBC

```
00000000 53 61 6C 74 65 64 5F 5F 33 B8 DA 16 1C B2 CC E1 Salted__3.....
00000010 D3 D3 0D 8B 2D 94 7B 14 E3 49 9C 39 E6 90 70 19 Y.j.....Y..a...
00000020 C9 7B 99 F6 F4 6E 86 5F B0 95 1A 24 A0 55 D5 2E E.{...n._...$.U..
00000030 35 AB B1 6C 50 6D FC 2C CC 01 B2 E2 7A 09 DA A3 5..lPm.,....z...
00000040 B1 F2 8A E3 BE 6F 7A 4F 68 1F 6B CF E9 25 28 CA .....oz0h.k..%(.
00000050 98 29 04 F2 80 C4 BB 98 E5 D0 D9 42 1D 22 00 AE .).....B."...
00000060 0C 4F 06 E1 1C 5C 67 A6 DB FD 4D 96 24 ED 74 66 .0....\g...M.$.tf
00000070 20 64 3D B7 78 79 4B 15 0D A2 8E C2 E9 B7 7C FD d=.xyK.....|.
00000080 A5 FD D4 A1 36 FC 6F DA 8E CF C5 04 E5 62 D2 07 ....6.o.....b..
00000090 50 84 08 B1 12 5D 1B 04 95 B4 F2 D5 80 A1 50 37 P....].....P7
```

Figura 4.5: input AES-128 modo CBC

```
00000000 53 61 6C 74 65 64 5F 07 6B 0A F5 64 F8 61 BF Salted__k..d.a.
00000010 D3 D3 0D 8B 2D 94 7B 14 E3 49 9C 39 E6 90 70 19 .....-.{...I.9..p.
00000020 92 36 B2 21 F1 E8 5A 8A 21 F5 14 36 CB 76 8F 12 .6.!...Z.!...6.v..
00000030 74 60 59 4A 6A A4 A7 C0 AB B0 E3 B4 0A 70 C5 09 t`YJj.....p..
00000040 34 05 34 EC 5D 9F 23 CC 52 03 9C 46 18 96 5F 85 4.4.]#.#.R..F...
00000050 B9 A6 26 A9 DB FE 9C FA BF 8A F5 ED 90 58 B1 6B .&.....X.k
00000060 91 23 0A 89 B3 F8 31 AC 0C AB 5F F6 41 EF 4A 1E .#....1..._A.J.
00000070 28 01 51 BB 00 7F 23 AB E6 43 DE 69 8D 83 2F EA (.Q...#..C.i../.
00000080 F8 7C 6D 33 41 87 86 96 0A 3E 3D 40 C2 85 EC 35 .|m3A....>=@...5
00000090 A8 F4 41 1D 99 D5 C3 79 40 C3 94 38 1A B8 03 3B .A....y@..8...;
```

Figura 4.6: input1 AES-128 modo CBC

## 4.3. Cifrado AES-128 modo OFB

Para cifrar con AES-128 en modo OFB con contraseña usamos el siguiente comando.

```
patri@patri:~/Escritorio$ openssl enc -aes-128-ofb -in input1.bin -pass pass:123456 -out input1ofb_aes128.enc
```

Figura 4.7: Comando para cifrar con AES-128 modo OFB



00000000	53 61 6C 74 65 64 5F 5F 4A 1D A2 C9 E6 E3 78 87	Salted_J....x.
00000010	1D 3E BD 33 E5 C6 64 09 76 0A 35 83 5C 0F 90 25	.>.3..d.v.5.\..%
00000020	87 0C 8D 2E A5 74 9F 27 B7 00 69 8C 02 2F B3 1F	.....t.'...i../..
00000030	20 75 4C 16 1D 02 F5 4A 0F 3A 31 E3 CB D9 4E 17	uL....J.:1..N.
00000040	79 A2 35 CB 17 EF 22 2E E6 EC C6 5D 94 D8 F2 59	y.5..."....]...Y
00000050	CA 60 7A C0 96 6E 22 DA C5 59 F1 38 73 14 61 23	.`z..n"....Y.8s.a#
00000060	F0 E4 03 DD D2 5A BD B5 02 CB B1 06 EC 00 D8 72	.....Z.....r
00000070	07 C7 B4 96 BE 38 62 98 FD 98 10 21 3B C8 DA 32	.....8b.....!;..2
00000080	E7 C8 7C F5 85 33 3C 42 29 88 2D 97 46 45 FE 15	... .3<B)..-.FE..

Figura 4.8: input AES-128 modo OFB

00000000	53 61 6C 74 65 64 5F 5F B6 E4 22 1D 4A D0 8F 96	Salted_..."J...
00000010	F9 4E 83 EB DD 85 5B 41 8D 6A D2 C9 EE 59 D9 97	.N....[A.j...Y..
00000020	B8 7E 97 63 E8 F7 B4 31 83 C0 1A D8 31 25 EF 35	.~.c...1...1%.5
00000030	9B AF 09 A9 D7 78 56 59 07 CF 2A 48 3E 78 3A 50	.....xVY...*H>x:P
00000040	1F 25 78 94 B6 CD 11 25 A5 0A CC C3 6C EF EF C4	.%x....%...l...
00000050	BF C6 2C 33 01 FC 49 9B 96 1A 64 3B 46 C4 07 7E	...3..I...d;F..~
00000060	25 E2 4F 1F 4F 26 B5 C0 D8 CC 0A 13 FA 99 AB C4	%..0.0&.....
00000070	0E 3E 87 7E 57 0C 1D B1 8B C5 17 E3 37 A0 3E C3	.>..~W.....7.>.
00000080	1A 14 B4 D7 08 69 E7 71 D9 6D 4D B1 31 3A D5 85	.....i.q.mM.1:..

Figura 4.9: input1 AES-128 modo OFB

En este caso hemos utilizado los mismo modos de cifrado que en el punto anterior, ECB, CBC y OFB, por lo tanto la forma de cifrar es la misma, con la salvedad de que en este caso no hemos usado clave, si no contraseña. Al usar contraseña, la clave y vector se obtienen a partir de la contraseña mediante funciones hash. La diferencia que se puede apreciar al usar contraseña es la cadena que aparece en la salida "salted", por defecto al introducir el comando usando contraseña añade la opción -salt una cadena aleatoria que agrega openssl a la contraseña para derivar la clave. Por eso podemos observar un bloque más para la cadena aleatoria. Esta opción evita ciertos ataques de diccionario.

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

### 5.1. Cifrado AES-128 modo ECB opción -nosalt

Para cifrar con AES-128 en modo ECB con contraseña usando -nosalt usamos el siguiente comando.

```
patri@patri:~/Escritorio$ openssl enc -aes-128-ecb -in input1.bin -pass pass:123456 -out input1ecb_aes128_nosalt.enc -nosalt
```

Figura 5.1: Comando para cifrar con AES-128 modo ECB -nosalt

```

00000000 6F 1F 38 68 E0 80 C2 0B 2D AD F2 F2 79 30 8E 93 0.8h....-...y0..
00000010 6F 1F 38 68 E0 80 C2 0B 2D AD F2 F2 79 30 8E 93 0.8h....-...y0..
00000020 6F 1F 38 68 E0 80 C2 0B 2D AD F2 F2 79 30 8E 93 0.8h....-...y0..
00000030 6F 1F 38 68 E0 80 C2 0B 2D AD F2 F2 79 30 8E 93 0.8h....-...y0..
00000040 6F 1F 38 68 E0 80 C2 0B 2D AD F2 F2 79 30 8E 93 0.8h....-...y0..
00000050 6F 1F 38 68 E0 80 C2 0B 2D AD F2 F2 79 30 8E 93 0.8h....-...y0..
00000060 6F 1F 38 68 E0 80 C2 0B 2D AD F2 F2 79 30 8E 93 0.8h....-...y0..
00000070 6F 1F 38 68 E0 80 C2 0B 2D AD F2 F2 79 30 8E 93 0.8h....-...y0..
00000080 B2 BC A3 19 11 79 39 46 2D AE 56 5F F9 7B B9 C4 .....y9F-.V_.{..

```

Figura 5.2: input AES-128 modo ECB -nosalt

```

00000000 6F 1F 38 68 E0 80 C2 0B 2D AD F2 F2 79 30 8E 93 0.8h....-...y0..
00000010 D9 B9 C9 1C 3B AF 42 3C 5E 86 53 25 20 26 BC 66 .....;B<^S% &.f
00000020 6F 1F 38 68 E0 80 C2 0B 2D AD F2 F2 79 30 8E 93 0.8h....-...y0..
00000030 6F 1F 38 68 E0 80 C2 0B 2D AD F2 F2 79 30 8E 93 0.8h....-...y0..
00000040 6F 1F 38 68 E0 80 C2 0B 2D AD F2 F2 79 30 8E 93 0.8h....-...y0..
00000050 6F 1F 38 68 E0 80 C2 0B 2D AD F2 F2 79 30 8E 93 0.8h....-...y0..
00000060 6F 1F 38 68 E0 80 C2 0B 2D AD F2 F2 79 30 8E 93 0.8h....-...y0..
00000070 6F 1F 38 68 E0 80 C2 0B 2D AD F2 F2 79 30 8E 93 0.8h....-...y0..
00000080 B2 BC A3 19 11 79 39 46 2D AE 56 5F F9 7B B9 C4 .....y9F-.V_.{..

```

Figura 5.3: input1 AES-128 modo ECB -nosalt

## 5.2. Cifrado AES-128 modo CBC opción -nosalt

Para cifrar con AES-128 en modo CBC con contraseña usando -no salt usamos el siguiente comando.

```

patri@patri:~/Escritorio$ openssl enc -aes-128-cbc -in input.bin -pass pass:1234
56 -out inputcbc_aes128_nosalt.enc -nosalt

```

Figura 5.4: Comando para cifrar con AES-128 modo CBC -nosalt



00000000	0A F1 D0 39 7E 67 7D 5F 6E 5F 09 AF E9 52 97 78	...9~g}_n_...R.x
00000010	FF 6F 64 02 2E 9A 1D 73 81 AA 0B 9A 11 0E C3 5E	.od....s.....^
00000020	84 22 A0 D2 47 EA DE 08 3E 52 B1 FB 64 84 36 39	..."G...>R..d.69
00000030	6E 58 80 F7 A5 62 31 6B 9B 66 C7 36 79 31 05 7F	nX...b1k.f.6y1..
00000040	66 FE 34 7C 11 D4 D7 48 81 3B EF E7 2B 53 FF D1	f.4 ...H.;...+S..
00000050	F3 A4 4D D3 DC F3 F9 96 EE F0 27 F5 A0 D0 20 5A	..M.....'... Z
00000060	CB 30 F0 38 10 71 6D 98 AE EF BF B6 26 91 D3 51	.0.8.qm.....&..Q
00000070	E4 BD 4B 08 FD BD 58 17 85 62 85 20 F1 9A 5B ED	..K...X..b. ...[.
00000080	DE B6 8D EE B6 DE 61 34 05 BF BC C9 3B A1 18 ED	.....a4....;

Figura 5.5: input AES-128 modo CBC -nosalt

00000000	0A F1 D0 39 7E 67 7D 5F 6E 5F 09 AF E9 52 97 78	...9~g}_n_...R.x
00000010	D2 26 DC 6C 35 91 22 A4 7B 39 2B 88 E4 D8 B9 AE	..&.15.".{9+....
00000020	EA 01 EE 84 89 A6 1D FD 08 8E 8D C3 D2 60 DC 50	.....`..P
00000030	A5 85 5C 80 AC 08 59 F5 E9 DB CD 7E 89 81 84 4D	...Y....~...M
00000040	83 BD E0 1E 93 10 9B 79 AD 5E F8 73 2E 25 60 4F	.....y.^..s.%`0
00000050	44 F1 E5 5B 79 0E 62 25 8C 6C 96 DD 90 21 2D BF	D..[y.b%.l...!-.
00000060	D0 5A C3 A5 72 6E 98 D1 83 94 C0 22 BA BD 46 28	.Z..rn....."..F(
00000070	4C 93 4E 1C E1 06 4C E8 C7 AA 1A 70 F0 0C 7D DD	L.N...L....p..}.
00000080	3C F7 B1 41 94 77 EA 9C 14 57 B5 E3 5B 46 CD 9D	<...A.w...W..[F..

Figura 5.6: input1 AES-128 modo CBC -nosalt

### 5.3. Cifrado AES-128 modo OFB opción -nosalt

Para cifrar con AES-128 en modo OFB con contraseña usando -no salt usamos el siguiente comando.

```
patri@patri:~/Escritorio$ openssl enc -aes-128-ofb -in input.bin -pass pass:1234
56 -out inputofb_aes128_nosalt.enc -nosalt
```

Figura 5.7: Comando para cifrar con AES-128 modo OFB -nosalt

```

00000000 0A F1 D0 39 7E 67 7D 5F 6E 5F 09 AF E9 52 97 78 |..9~g}_n...R.x
00000010 FF 6F 64 02 2E 9A 1D 73 81 AA 0B 9A 11 0E C3 5E |.od....s.....^
00000020 84 22 A0 D2 47 EA DE 08 3E 52 B1 FB 64 84 36 39 |"..G...>R..d.69
00000030 6E 58 80 F7 A5 62 31 6B 9B 66 C7 36 79 31 05 7F |nX...b1k.f.6y1..
00000040 66 FE 34 7C 11 D4 D7 48 81 3B EF E7 2B 53 FF D1 |f.4|...H.;...+S..
00000050 F3 A4 4D D3 DC F3 F9 96 EE F0 27 F5 A0 D0 20 5A |..M.....'... Z
00000060 CB 30 F0 38 10 71 6D 98 AE EF BF B6 26 91 D3 51 |.0.8.qm....&..Q
00000070 E4 BD 4B 08 FD BD 58 17 85 62 85 20 F1 9A 5B ED |..K...X..b. ..[.

```

Figura 5.8: input AES-128 modo OFB -nosalt

```

00000000 0A F1 D0 39 7E 67 7D 5F 6E 5F 09 AF E9 52 97 78 |..9~g}_n...R.x
00000010 FF 7F 64 02 2E 9A 1D 73 81 AA 0B 9A 11 0E C3 5E |..d....s.....^
00000020 84 22 A0 D2 47 EA DE 08 3E 52 B1 FB 64 84 36 39 |"..G...>R..d.69
00000030 6E 58 80 F7 A5 62 31 6B 9B 66 C7 36 79 31 05 7F |nX...b1k.f.6y1..
00000040 66 FE 34 7C 11 D4 D7 48 81 3B EF E7 2B 53 FF D1 |f.4|...H.;...+S..
00000050 F3 A4 4D D3 DC F3 F9 96 EE F0 27 F5 A0 D0 20 5A |..M.....'... Z
00000060 CB 30 F0 38 10 71 6D 98 AE EF BF B6 26 91 D3 51 |.0.8.qm....&..Q
00000070 E4 BD 4B 08 FD BD 58 17 85 62 85 20 F1 9A 5B ED |..K...X..b. ..[.

```

Figura 5.9: input1 AES-128 modo OFB -nosalt

En los tres modos al usar el argumento -nosalt vemos como en la salida no está la palabra salted,y observamos un bloque menos. Además de que con esta opción, la misma contraseña siempre genera la misma clave de cifrado. Es posible realizar ataques de diccionario eficientes contra la contraseña y atacar datos cifrados de flujo de datos.

## 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

El cifrado AES-192 varía con respecto al AES-256 en el tamaño de clave, que en este caso es de 192 bits.

```

patri@patri:~/Escritorio$ openssl enc -aes-192-ofb -in input.bin -K 123456789012345678
9012345678901234567890484848 -iv 0123456789abcdef1372393311abcdef -out output.bin
patri@patri:~/Escritorio$

```

Figura 6.1: Comando para cifrar con AES-192 modo OFB

```

00000000 5A 01 1C BA 98 58 32 46 3B 24 22 F1 45 4D E4 80 Z....X2F;$".EM..
00000010 F0 28 45 D0 C9 C7 A5 D9 97 E8 F6 ED 46 41 B2 72 .(E.....FA.r
00000020 FA 52 48 25 A7 32 7C C4 2E 9B 81 30 37 46 5B 11 .RH%.2|....07F[.
00000030 69 8E C8 B0 F3 3B 84 E0 01 B3 F8 76 83 25 A8 7F i....;....v.%..
00000040 A0 EC 70 51 B0 D5 BA 29 1C 6B DF 1C 9A 1B 23 63 .pQ...).k...#c
00000050 8A A0 AA 39 E1 97 43 81 9A 39 DC 6F 9B EB B3 86 ...9..C..9.o....
00000060 3C 90 F3 C1 BB 55 D3 DF 24 42 4F 89 5F 3A 6C AC <....U..$B0._:l.
00000070 04 A7 C6 6E E0 B4 00 5B 0B 23 8A C5 4B B0 E5 D8 ...n...[.#..K...

```

Figura 6.2: input AES-192 modo OFB

## 7. Descifrar output.bin utilizando a misma clave y vector de inicialización que en 6.

Para descifrar usamos la opción -d, además de la clave y vector con el que ciframos el archivo.

```

patri@patri:~/Escritorio$ openssl enc -aes-192-ofb -d -in input.bin -K 123456789012345
678901234567890123456789048484848 -iv 0123456789abcdef1372393311abcdef -out output_des
crypted
.bin

```

Figura 7.1: Comando para descifrar con AES-192 modo OFB

```

00000000 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00000010 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00000020 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00000030 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00000040 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00000050 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00000060 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00000070 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....

```

Figura 7.2: output.bin descifrado AES-192 modo OFB

Al descifrar el archivo output.bin con la misma clave y vector obtenemos el original contenido con ceros.

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**

```
patri@patri:~/Escritorio$ openssl enc -aes-192-ofb -in output.bin -K 123456789012345678901234567890123456789048484848 -iv 0123456789abcdef1372393311abcdef -out output_cifrado.bin
```

Figura 8.1: Comando para cifrar con AES-192 modo OFB



A hex dump of the file output.bin. The first 8 lines show addresses from 00000000 to 00000070. Each line contains 32 hex characters, all of which are '00'. The last character of each line is followed by a period and a space. The dump indicates that the entire file consists of null bytes (zeros).

Figura 8.2: output.bin cifrado AES-192 modo OFB

Al cifrar dos veces con el modo OFB lo que conseguimos es volver a descifrar, por eso hemos obtenido el mismo resultado que en el punto 7, el archivo relleno de ceros. Le hemos pasado un archivo ya cifrado y obtenemos el resultado contrario a cifrar, esto ocurre por la forma la que actúa el modo OFB que el cifrado y descifrado lo hace de la misma manera.

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

Cifrad input.bin con AES-192 en modo OFB.

```
patri@patri:~/Escritorio$ openssl enc -aes-192-ofb -in input.bin -pass pass:1234567 -out output2.bin
```

Figura 9.1: Comando para cifrar con AES-192 modo OFB con contraseña

```

00000000 53 61 6C 74 65 64 5F 5F 25 EF 85 48 01 F4 AB 79 Salted_%.H...y
00000010 9B E4 3A C1 68 69 F0 60 23 35 02 82 2C 01 F7 BF ...hi.`#5....
00000020 06 4A 6F B8 FE 41 AA 27 0B 6F D2 5B 04 D7 54 F1 .Jo..A.'o.[..T.
00000030 D9 01 38 CF 35 CC 6C 0F 12 A5 D0 1E 82 98 7E 89 ..8.5.l.....~.
00000040 15 2E 91 21 D8 F8 27 AD 5E E5 D8 0F 8A D7 30 B0 ...!...' ^.....0.
00000050 71 93 5F A6 B0 D6 D9 1A 2D 8F BA A8 BE 8E 5D 4C q.....]L
00000060 92 63 3A 55 73 4F 9F 9F A4 41 90 B6 DA 3A 7D 4C .c:Us0...A...}L
00000070 25 5F 6C BF 48 C3 F3 BA 27 4C 06 58 A1 2E 05 A8 %_l.H...'L.X...
00000080 1F AE 39 55 7D B0 B8 E6 B3 72 45 20 40 3B AC D4 ..9U}....rE @;..

```

Figura 9.2: input.bin cifrado AES-192 modo OFB con contraseña

Descifrad output.bin.

```

patri@patri:~/Escritorio$ openssl enc -aes-192-ofb -d -in output2.bin -pass pass:1234567 -out output2_des.bin

```

Figura 9.3: Comando para descifrar con AES-192 modo OFB con contraseña

```

00000000 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00000010 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00000020 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00000030 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00000040 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00000050 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00000060 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00000070 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....

```

Figura 9.4: output2.bin descifrado AES-192 modo OFB con contraseña

Vuelve a cifrar output2.bin con AES-192 en modo OFB.

```

patri@patri:~/Escritorio$ openssl enc -aes-192-ofb -in output2.bin -pass pass:1234567 -out output2_cifrado.bin

```

Figura 9.5: Comando para cifrar con AES-192 modo OFB con contraseña

```

00000000 5B 61 6C 74 65 64 5F 5F 51 B2 92 F6 7C 97 C0 2B Salted_Q...|...+
00000010 CC E6 B9 F7 8C 84 E6 31 9B 91 12 5F F2 96 DA 39 .....1..._...9
00000020 0B 6A 03 F7 27 4A C8 D1 D1 21 11 84 67 FC 1B F3 .j..'J...!.g...
00000030 00 A7 6C 9A E4 93 EF 38 6C 34 06 84 2C 67 5F B6 ..l....8l4...g_.
00000040 A0 7A 90 EA 4E 06 19 C0 3E D0 41 20 82 2A 8F 74 .z..N...>.A .*t
00000050 B3 3C 6C 1B 6D 1C 75 97 67 FE C1 A9 93 5E F3 1A .<l.m.u.g....^..
00000060 15 F3 9C DF F9 88 4E 91 E2 39 74 98 76 AF 39 C5 .....N..9t.v.9.
00000070 11 CD 9E 02 A5 B7 F9 C6 C5 91 E3 04 C5 E9 27 68 ..... 'h
00000080 CB EC 3E 36 86 24 80 3D BC A1 91 B5 C2 7F A0 DA ..>6.$.=.....
00000090 F7 56 89 AF A5 B1 C4 F8 2C 13 76 82 7B 55 DF 2B .V.....,v.{U.+

```

Figura 9.6: output2.bin cifrado AES-192 modo OFB con contraseña

Comprobamos que no ocurre como en el punto anterior en el que usábamos clave y vector, ya que lo que conseguimos en este caso es volver a cifrar. Esto ocurre porque con contraseña añade la opción -salt por defecto añade una cadena aleatoria a la contraseña para derivar la clave como comentamos anteriormente. Es por esto por lo que al volver a cifrar no contempla el bloque añadido de salto, ya que estamos cifrando y lo asimila como un bloque más, por esto no podemos volver al archivo original que contenía ceros. Además al meter otro salt al cifrar con contraseña, aunque esta contraseña sea la misma, la clave ya no es la misma.

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

Algoritmos de mi implementación de OpenSSL:

```

Cipher commands (see the 'enc' command for more details)
aes-128-cbc      aes-128-ecb      aes-192-cbc      aes-192-ecb
aes-256-cbc      aes-256-ecb      base64           bf
bf-cbc          bf-cfb          bf-ecb           bf-ofb
camellia-128-cbc camellia-128-ecb camellia-192-cbc camellia-192-ecb
camellia-256-cbc camellia-256-ecb cast             cast-cbc
cast5-cbc       cast5-cfb       cast5-ecb       cast5-ofb
des             des-cbc        des-cfb        des-ecb
des-ede        des-ede-cbc    des-ede-cfb    des-ede-ofb
des-ede3       des-ede3-cbc   des-ede3-cfb   des-ede3-ofb
des-ofb        des3           desx           rc2
rc2-40-cbc     rc2-64-cbc     rc2-cbc        rc2-cfb
rc2-ecb       rc2-ofb        rc4            rc4-40
seed          seed-cbc       seed-cfb       seed-ecb
seed-ofb

```

Figura 10.1: Algoritmos de mi implementación de OpenSSL

En mi caso he elegido **Camelia**, es un cifrado simétrico de bloque de clave con un tamaño de bloque de 128 bits y tamaños de 128, 192 y 256 bits. El cifrado tiene niveles de seguridad y capacidades de procesamiento comparables con AES. Es un cifrado Feistel

con 18 rondas (cuando se usan claves de 128 bits) o 24 rondas (cuando se usan claves de 192 o 256 bits). Cada seis rondas, se aplica una capa de transformación lógica: la llamada "función FL.<sup>o</sup> su inversa. Camellia utiliza cuatro cajas S de 8x8 bits con transformaciones afines de entrada y salida y operaciones lógicas.

Camelia se considera un cifrado seguro. Incluso si se usa la opción de tamaño de clave más pequeña de 128 bits, se considera que no es posible romperla mediante un ataque de fuerza bruta sobre las teclas con la tecnología actual.

## **11. Repetid los puntos de 3 a 5 con el cifrado presentado en el punto 10.**

Para este punto se va a utilizar el cifrado Camelia-128 con clave de 128 bits. En los modos siguientes vamos a comprobar como funcionan como hemos visto anteriormente en AES, aunque cambiemos de cifrado los modos se comportan igual.

### **11.1. Cifrado input.bin e input1.bin con Camellia-128 en modos ECB, CBC y OFB usando una clave.**

#### **11.1.1. Cifrado Camellia-128 modo ECB**

Comando utilizado para cifrar con Camellia-128 modo ECB:

```
patri@patri:~/Escritorio$ openssl camellia-128-ecb -K 23232323232323232323232323232323232312 -in input.bin -out o_camelia_ecb.bin
```

Figura 11.1: Comando Camellia-128 modo ECB

00000000	9A EA 02 27 7F 92 65 18 5A 16 0E E2 7C A9 83 EC	...'.e.Z... ...
00000010	9A EA 02 27 7F 92 65 18 5A 16 0E E2 7C A9 83 EC	...'.e.Z... ...
00000020	9A EA 02 27 7F 92 65 18 5A 16 0E E2 7C A9 83 EC	...'.e.Z... ...
00000030	9A EA 02 27 7F 92 65 18 5A 16 0E E2 7C A9 83 EC	...'.e.Z... ...
00000040	9A EA 02 27 7F 92 65 18 5A 16 0E E2 7C A9 83 EC	...'.e.Z... ...
00000050	9A EA 02 27 7F 92 65 18 5A 16 0E E2 7C A9 83 EC	...'.e.Z... ...
00000060	9A EA 02 27 7F 92 65 18 5A 16 0E E2 7C A9 83 EC	...'.e.Z... ...
00000070	9A EA 02 27 7F 92 65 18 5A 16 0E E2 7C A9 83 EC	...'.e.Z... ...
00000080	DE DE 7B 5C 44 E3 3E 8A 1B 2D 81 74 F6 01 07 60	...{\D.>...t...`

Figura 11.2: input camellia-128 modo ECB

00000000	9A EA 02 27 7F 92 65 18 5A 16 0E E2 7C A9 83 EC	...'.e.Z... ...
00000010	B5 E2 FD 14 FE CA 06 3F DC C6 84 4F F0 09 A0 64	.....?.0...d
00000020	9A EA 02 27 7F 92 65 18 5A 16 0E E2 7C A9 83 EC	...'.e.Z... ...
00000030	9A EA 02 27 7F 92 65 18 5A 16 0E E2 7C A9 83 EC	...'.e.Z... ...
00000040	9A EA 02 27 7F 92 65 18 5A 16 0E E2 7C A9 83 EC	...'.e.Z... ...
00000050	9A EA 02 27 7F 92 65 18 5A 16 0E E2 7C A9 83 EC	...'.e.Z... ...
00000060	9A EA 02 27 7F 92 65 18 5A 16 0E E2 7C A9 83 EC	...'.e.Z... ...
00000070	9A EA 02 27 7F 92 65 18 5A 16 0E E2 7C A9 83 EC	...'.e.Z... ...
00000080	DE DE 7B 5C 44 E3 3E 8A 1B 2D 81 74 F6 01 07 60	...{\D.>...t...`

Figura 11.3: input1 camellia-128 modo ECB

En input podemos comprobar como pasa como en el cifrado aes con modo ECB, ya que todos los bloques de 128 bits se repiten al estar todos rellenos de ceros, menos el último boque que corresponde al padding. En input2 cambia el segundo bloque donde cambiamos el bit a 1 y también tiene padding.

### 11.1.2. Cifrado Camelia-128 modo CBC

Comando utilizado para cifrar con Camelia-128 modo CBC:

```
patri@patri:~/Escritorio$ openssl camellia-128-cbc -K 23232323232323232323232323232323232312 -iv 0123456789abcdef0123456789abcdef -in input.bin -out o_camelia_cbc.bin
```

Figura 11.4: Comando Camelia-128 modo CBC



00000000	11	8B 4E 98 46 E1 5E 40 E0 B5 70 97 0E 9F 57 18	.N.F.^@..p...W.
00000010	9C	D9 F9 F5 6C C2 F3 62 1B C5 E6 07 B6 68 49 44	....l..b....hID
00000020	03	0E 82 BF 25 C7 31 0A EF 75 8F CC DE 13 1D 2B	....%.1..u....+
00000030	5F	00 E0 4D 8F FC 63 92 E3 3E 46 FD 76 DE CF DC	_..M..c..>F.v...
00000040	41	48 3F 51 9A 7E DE 71 CA 64 C4 79 84 73 F0 90	AH?Q.~.q.d.y.s..
00000050	D0	0D FC 51 FA EA 24 8E 8A EA CC 16 D5 81 30 62	...Q...\$.....0b
00000060	89	C0 28 9C 87 92 12 E2 B8 9A 9B 39 E0 BF 0D 08	..(.....9....
00000070	24	13 F9 DF DF 2E 47 44 4B 3B F2 CF D0 8E F2 3B	\$.....GDK;.....;
00000080	95	59 E1 CB 2B 35 7B 1D 6A 32 65 07 09 C1 89 49	.Y...+5{.j2e....I

Figura 11.5: input camellia-128 modo CBC

00000000	11 8B 4E 98 46 E1 5E 40 E0 B5 70 97 0E 9F 57 18	.N.F.^@.p...W.
00000010	CB 3A C2 F2 0D 93 A4 B1 3A 4A A3 24 DC A1 20 49	.....:J.\$.. I
00000020	C9 1E B8 F8 13 D8 AE 07 52 B5 B0 E4 99 FA 1B CB	.....R.....
00000030	A5 A8 88 B3 77 A8 0F 84 6B 24 D9 A4 7C 49 4A 1D	...w....k\$. IJ.
00000040	BC 3A 8F 1D 82 06 8F 14 C7 39 28 92 11 76 84 9B	.....9(..v..
00000050	34 71 37 29 88 CB E2 C2 2E EC 5F 67 59 EB 88 F7	4q7)....._gY...
00000060	59 3D C6 5F 33 59 53 17 97 4B 88 07 0C 84 1A 89	Y=._3YS..K.....
00000070	B8 6B FA B6 59 8C 77 32 FF 1F EF 5B 04 43 48 D2	.k..Y.w2...[.CH.
00000080	65 05 79 8E 2D F0 1D BC B8 7D E2 16 23 85 3A 38	e.y.-....}.#. :8

Figura 11.6: input1 camellia-128 modo CBC

Se va encadenando el cifrado anterior con el nuevo cifrado usandolo como vector de suma XOR para el nuevo bloque de texto a cifrar. Al depender cada bloque de todos los bloques anteriores podemos observar como de input a input1, en input1 cambia a partir del bloque en el que se cambio el bit a 1. Más el último bloque de padding en ambos.

### 11.1.3. Cifrado Camelia-128 modo OFB

Comando utilizado para cifrar con Camelia-128 modo OFB:

```
patri@patri:~/Escritorio$ openssl camellia-128-ofb -K 23232323232323232323232323232323
232312 -iv 0123456789abcdef0123456789abcdef -in input1.bin -out o1_camelia_ofb.b
in
```

Figura 11.7: Comando Camelia-128 modo OFB

```

00000000 11 8B 4E 98 46 E1 5E 40 E0 B5 70 97 0E 9F 57 18 .N.F.^@..p...W.
00000010 9C D9 F9 F5 6C C2 F3 62 1B C5 E6 07 B6 68 49 44 ....l..b....hID
00000020 03 0E 82 BF 25 C7 31 0A EF 75 8F CC DE 13 1D 2B ....%.1..u....+
00000030 5F 00 E0 4D 8F FC 63 92 E3 3E 46 FD 76 DE CF DC ..M..c..>F.v...
00000040 41 48 3F 51 9A 7E DE 71 CA 64 C4 79 84 73 F0 90 AH?Q.~.q.d.y.s..
00000050 D0 0D FC 51 FA EA 24 8E 8A EA CC 16 D5 81 30 62 ...Q..$......0b
00000060 89 C0 28 9C 87 92 12 E2 B8 9A 9B 39 E0 BF 0D 08 ..(.....9....
00000070 24 13 F9 DF DF 2E 47 44 4B 3B F2 CF D0 8E F2 3B $.....GDK;.....;

```

Figura 11.8: input camellia-128 modo OFB

```

00000000 11 8B 4E 98 46 E1 5E 40 E0 B5 70 97 0E 9F 57 18 .N.F.^@..p...W.
00000010 9C C9 F9 F5 6C C2 F3 62 1B C5 E6 07 B6 68 49 44 ....l..b....hID
00000020 03 0E 82 BF 25 C7 31 0A EF 75 8F CC DE 13 1D 2B ....%.1..u....+
00000030 5F 00 E0 4D 8F FC 63 92 E3 3E 46 FD 76 DE CF DC ..M..c..>F.v...
00000040 41 48 3F 51 9A 7E DE 71 CA 64 C4 79 84 73 F0 90 AH?Q.~.q.d.y.s..
00000050 D0 0D FC 51 FA EA 24 8E 8A EA CC 16 D5 81 30 62 ...Q..$......0b
00000060 89 C0 28 9C 87 92 12 E2 B8 9A 9B 39 E0 BF 0D 08 ..(.....9....
00000070 24 13 F9 DF DF 2E 47 44 4B 3B F2 CF D0 8E F2 3B $.....GDK;.....;

```

Figura 11.9: input1 camellia-128 modo OFB

Comprobamos como al cambiar un bit de input a input1, cambia solo la posición en la que tenemos ese bit, y el resto de bloques son exactamente igual. En este modo como sabemos tenemos un bloque menos ya que no necesita padding.

## 11.2. Cifrado input.bin e input1.bin con Camellia-128 en modos ECB, CBC y OFB usando contraseña.

### 11.2.1. Cifrado Camelia-128 modo ECB con contraseña

Comando utilizado para cifrar con Camelia-128 modo ECB con contraseña:

```

patri@patri:~/Escritorio$ openssl camellia-128-ecb -pass pass:123456 -in
input.bin -out o_camelia_ecb_salt.bin

```

Figura 11.10: Comando Camelia-128 modo ECB con contraseña

```

00000000 53 61 6C 74 65 64 5F 5F 14 0F 93 2E 87 76 BA 75 Salted____.v.u
00000010 2B 5F 43 17 D1 0B 6C 35 B0 F4 36 9A FD C9 44 50 +_C...l5..6...DP
00000020 2B 5F 43 17 D1 0B 6C 35 B0 F4 36 9A FD C9 44 50 +_C...l5..6...DP
00000030 2B 5F 43 17 D1 0B 6C 35 B0 F4 36 9A FD C9 44 50 +_C...l5..6...DP
00000040 2B 5F 43 17 D1 0B 6C 35 B0 F4 36 9A FD C9 44 50 +_C...l5..6...DP
00000050 2B 5F 43 17 D1 0B 6C 35 B0 F4 36 9A FD C9 44 50 +_C...l5..6...DP
00000060 2B 5F 43 17 D1 0B 6C 35 B0 F4 36 9A FD C9 44 50 +_C...l5..6...DP
00000070 2B 5F 43 17 D1 0B 6C 35 B0 F4 36 9A FD C9 44 50 +_C...l5..6...DP
00000080 2B 5F 43 17 D1 0B 6C 35 B0 F4 36 9A FD C9 44 50 +_C...l5..6...DP
00000090 14 8D B7 E9 4A 2F 98 09 38 9F 38 60 75 67 55 51 .....J/..8.8`ugUQ

```

Figura 11.11: input camellia-128 modo ECB con contraseña

```

00000000 53 61 6C 74 65 64 5F 5F E6 BE 01 04 E9 FD 43 5A Salted____.CZ
00000010 BD 72 23 CE CF EF 5C C0 F9 66 FE 1C EA 62 20 F3 .r#...\.f...b .
00000020 7D D1 49 87 F9 AA 93 18 74 14 07 F5 99 9F DF 77 }.I.....t.....w
00000030 BD 72 23 CE CF EF 5C C0 F9 66 FE 1C EA 62 20 F3 .r#...\.f...b .
00000040 BD 72 23 CE CF EF 5C C0 F9 66 FE 1C EA 62 20 F3 .r#...\.f...b .
00000050 BD 72 23 CE CF EF 5C C0 F9 66 FE 1C EA 62 20 F3 .r#...\.f...b .
00000060 BD 72 23 CE CF EF 5C C0 F9 66 FE 1C EA 62 20 F3 .r#...\.f...b .
00000070 BD 72 23 CE CF EF 5C C0 F9 66 FE 1C EA 62 20 F3 .r#...\.f...b .
00000080 BD 72 23 CE CF EF 5C C0 F9 66 FE 1C EA 62 20 F3 .r#...\.f...b .
00000090 AC 0D A0 A2 93 0C E5 2C 19 23 03 EE 22 D2 CE EA .....,.#..."...

```

Figura 11.12: input1 camellia-128 modo ECB contraseña

En este caso hemos utilizado el cifrado con contraseña, como se ha dicho anteriormente, por defecto incluye la opción -salt, podemos observar que al igual que en cifrado aes, se añade un bloque más debido a la funcionalidad de -salt, añadiendo una cadena aleatoria a la contraseña para derivar la clave. En este y en los diferentes modos siguientes pasa igual que en el apartado 4 explicado anteriormente lo único que cambia es que hemos utilizado otro algoritmo de cifrado, pero en la salida podemos apreciar que se comporta de la misma forma.

### 11.2.2. Cifrado Camelia-128 modo CBC con contraseña

Comando utilizado para cifrar con Camelia-128 modo CBC con contraseña:

```

patri@patri:~/Escritorio$ openssl camellia-128-cbc -pass pass:123456 -in
input1.bin -out o1_camelia_cbc_salt.bin

```

Figura 11.13: Comando Camelia-128 modo CBC con contraseña

```

00000000 53 61 6C 74 65 64 5F 5F A7 A8 E0 04 E7 BC 13 C7 Salted____.
00000010 AD 3A 6B 8A FA BE 3B 83 5C 97 F2 F8 6E 2A 07 C0 .:k...;\...n*..
00000020 89 94 D7 B8 CF 82 0A 48 5F E8 4C 75 9B 70 E8 8A .....H_.Lu.p..
00000030 46 50 5B FC 98 BB 54 66 27 D5 13 00 E6 4A 8E 43 FP[...Tf'....J.C
00000040 37 EC 5F BB 08 8B A0 E4 DD 73 8F FF DC 56 5B CA 7_.....s...V[.
00000050 B0 B4 1C 28 90 BC C6 1E 6D BC 52 5E DE 38 98 F1 ...(.m.R^8..
00000060 92 E5 1A B9 03 23 78 28 69 A9 50 C9 5D 29 EE 90 .....#x(i.P.))..
00000070 B4 F9 97 B2 4A E0 A6 C6 DA 76 C4 CF B2 92 ED 5B ....J....v....[
00000080 B3 EE C4 60 E2 E8 13 61 FB 0F DD 7F 79 13 16 5A ...`...a....y..Z
00000090 6E 2E 12 AC 3E 2A EC A8 F5 53 E1 3C 49 A0 AA 57 n...>*...S.<I..W

```

Figura 11.14: input camellia-128 modo CBC con contraseña

```

00000000 53 61 6C 74 65 64 5F 5F DF 4D B5 AC EB 5B E4 32 Salted__M...[.2
00000010 67 79 B4 E0 CA 2E 80 E1 A4 88 25 91 60 68 37 3A gy.....%.`h7:
00000020 9A A8 25 B6 47 3C 3F 44 A6 06 14 D7 CA AE 20 45 ..%.G<?D..... E
00000030 46 B0 FC F1 6E 08 7A D1 AB C2 EC A1 27 7A BC 29 F...n.z.....'z.)
00000040 A5 3B E6 FE 85 22 E0 4B B5 AD FC E0 8B C0 52 58 .;...".K.....RX
00000050 03 A0 DB 7F C1 D4 6C 9F A7 01 AB C3 C0 7E 0D 96 .....l.....~..
00000060 E1 3D 37 BF 4A B6 52 D6 D1 8B 73 72 4A 50 3C 4A .=7.J.R...srJP<J
00000070 78 31 98 D5 1E F4 A0 70 16 73 FA B4 C7 52 F9 C3 x1....p.s...R..
00000080 DE B2 3B D9 1F 76 A2 C1 A4 23 E6 1C D3 D4 8E 87 .;...v...#.....
00000090 E0 9E 39 14 4C 03 91 E0 CA D1 C5 5A CD 55 A0 A7 ..9.L.....Z.U..

```

Figura 11.15: input1 camellia-128 modo CBC contraseña

### 11.2.3. Cifrado Camelia-128 modo OFB con contraseña

Comando utilizado para cifrar con Camelia-128 modo OFB con contraseña:

```

patri@patri:~/Escritorio$ openssl camellia-128-ofb -pass pass:123456 -in
input1.bin -out o1_camelia_ofb_salt.bin

```

Figura 11.16: Comando Camelia-128 modo OFB con contraseña

00000000	53	61	6C	74	65	64	5F	5F	65	7C	9B	5D	B5	3D	04	D4	Salted_e .].=..
00000010	A7	59	4C	66	40	33	DE	AA	02	8E	43	DB	96	C5	CC	86	.Ylf@3....C.....
00000020	83	22	9C	90	B3	66	C0	98	84	C2	17	0E	55	23	C7	1F	."...f.....U#..
00000030	01	64	45	6F	E5	96	25	4F	1A	EA	0F	81	31	F2	66	2E	.dEo..%0....1.f.
00000040	57	41	09	F0	65	66	1E	6A	3E	32	F5	4E	13	A2	09	84	WA..ef.j>2.N....
00000050	0F	4D	A7	63	C6	D4	06	4B	04	E3	5F	63	6B	8A	3F	03	.M.c...K...ck.?.
00000060	A6	DD	72	F4	60	EC	AA	F2	DC	CA	FD	19	E0	AA	40	E7	..r.`.....@.
00000070	B0	68	0F	77	42	1E	DC	7C	CC	A0	13	7E	C6	64	44	19	.h.wB.. ...~.dD.
00000080	F6	A6	19	CF	04	70	CD	F3	E2	40	3F	45	44	FD	A9	35	.....p...@?ED..5

Figura 11.17: input camellia-128 modo OFB con contraseña

00000000	53	61	6C	74	65	64	5F	5F	58	FA	19	75	7A	C7	30	C0	Salted_X..uz.0.
00000010	07	CF	2C	94	F6	7C	FF	77	48	EF	FB	AC	76	EF	A7	3C	.... .wH...v.<
00000020	B1	B5	59	C0	42	18	32	7E	CD	DF	8B	20	9B	2B	BA	33	..Y.B.2~...+.3
00000030	9E	21	B9	BE	73	2D	E0	A3	6A	C1	90	FB	40	FB	D9	CB	!.!s-..j...@...
00000040	41	97	59	27	3C	FD	38	6D	B4	4A	D4	7C	C9	4B	CF	C4	A.Y'<.8m.J. .K..
00000050	59	EB	BF	BE	27	73	67	0E	AD	7A	60	3C	43	53	E6	6F	Y...'sg..z`<CS.o
00000060	4C	7B	61	39	42	36	22	81	A4	E7	DF	4C	55	82	4F	19	L{a9B6"....LU.0.
00000070	D5	6B	52	B0	55	58	D9	63	99	CE	42	97	3A	E1	4C	93	.kR.UX.c..B.:.L.
00000080	AA	0E	DF	84	89	5B	4E	23	1C	3F	F2	79	6C	34	D4	1C	.....[N#.?..yl4..

Figura 11.18: input1 camellia-128 modo OFB contraseña

### 11.3. Repetición del punto anterior con la opción -nosalt

#### 11.3.1. Cifrado Camelia-128 modo ECB con contraseña opción -nosalt

Comando utilizado para cifrar con Camelia-128 modo ECB con contraseña opción -nosalt:

```
patri@patri:~/Escritorio$ openssl camellia-128-ecb -pass pass:123456 -in
input1.bin -out o1_camelia_ecb_nosalt.bin -nosalt
```

Figura 11.19: Comando Camelia-128 modo ECB con contraseña

```

00000000 54 4C B0 8A BA 7F DB 89 A1 9D AF 94 71 AE C3 F2 TL.....q...
00000010 54 4C B0 8A BA 7F DB 89 A1 9D AF 94 71 AE C3 F2 TL.....q...
00000020 54 4C B0 8A BA 7F DB 89 A1 9D AF 94 71 AE C3 F2 TL.....q...
00000030 54 4C B0 8A BA 7F DB 89 A1 9D AF 94 71 AE C3 F2 TL.....q...
00000040 54 4C B0 8A BA 7F DB 89 A1 9D AF 94 71 AE C3 F2 TL.....q...
00000050 54 4C B0 8A BA 7F DB 89 A1 9D AF 94 71 AE C3 F2 TL.....q...
00000060 54 4C B0 8A BA 7F DB 89 A1 9D AF 94 71 AE C3 F2 TL.....q...
00000070 54 4C B0 8A BA 7F DB 89 A1 9D AF 94 71 AE C3 F2 TL.....q...
00000080 8C 41 2D A9 EE 18 F8 00 31 CB 12 D3 0E B4 60 8D .A-.....1.....`

```

Figura 11.20: input camellia-128 modo ECB con contraseña opción -nosalt

```

00000000 54 4C B0 8A BA 7F DB 89 A1 9D AF 94 71 AE C3 F2 TL.....q...
00000010 80 9F 98 9B BC E8 E8 F6 07 CA C1 39 1F 2F 30 58 .....9./0X
00000020 54 4C B0 8A BA 7F DB 89 A1 9D AF 94 71 AE C3 F2 TL.....q...
00000030 54 4C B0 8A BA 7F DB 89 A1 9D AF 94 71 AE C3 F2 TL.....q...
00000040 54 4C B0 8A BA 7F DB 89 A1 9D AF 94 71 AE C3 F2 TL.....q...
00000050 54 4C B0 8A BA 7F DB 89 A1 9D AF 94 71 AE C3 F2 TL.....q...
00000060 54 4C B0 8A BA 7F DB 89 A1 9D AF 94 71 AE C3 F2 TL.....q...
00000070 54 4C B0 8A BA 7F DB 89 A1 9D AF 94 71 AE C3 F2 TL.....q...
00000080 8C 41 2D A9 EE 18 F8 00 31 CB 12 D3 0E B4 60 8D .A-.....1.....`

```

Figura 11.21: input1 camellia-128 modo ECB contraseña opción -nosalt

### 11.3.2. Cifrado Camelia-128 modo CBC con contraseña opción -nosalt

Comando utilizado para cifrar con Camelia-128 modo ECB con contraseña opción -nosalt:

```

patri@patri:~/Escritorio$ openssl camellia-128-ofb -pass pass:123456 -in
input1.bin -out o1_camellia_ofb_nosalt.bin -nosalt

```

Figura 11.22: Comando Camelia-128 modo OFB con contraseña

```

00000000 AC 86 26 6A 9C FF AC EC C4 90 6A 49 E3 C0 87 80 .&j.....jI...
00000010 39 08 DC 45 FA 5A CD 2A 79 8B 5C CC EF A0 A9 EA 9...E.Z.*y.\....
00000020 86 5C 19 C1 A6 42 8B 14 88 EB 41 1B 29 8B 4C 96 .\...B...A.)..L.
00000030 FC 64 4A 38 63 33 57 67 90 BF 7A 07 49 48 EA 33 .dJ8c3Wg...z.IH.3
00000040 8C 07 50 96 1E 14 E3 F8 E2 A7 55 9F E5 16 4E E9 .P.....U...N.
00000050 8C 5A BE E9 60 1B 37 7C B0 62 C2 A8 DA 4A F9 6B .Z...`.7|.b...J.k
00000060 FA 21 60 56 87 6B 66 9F 14 CA 1D 6A AD 12 8C 55 .!`V.kf....j...U
00000070 42 A8 B2 73 71 7C 89 43 15 77 EC AE D1 C8 05 F9 B..sq|.C.w.....
00000080 50 9F B4 D4 C4 01 80 9A 97 E9 45 0F 8B A8 E8 FDP.....E.....

```

Figura 11.23: input camellia-128 modo CBC con contraseña opción -nosalt

```

00000000 AC 86 26 6A 9C FF AC EC C4 90 6A 49 E3 C0 87 80 .&j.....jI...
00000010 BC 9F 72 25 CF 2E 4F 0C 7F A9 D5 E8 02 9D BB C0 ...r%..0.....
00000020 30 12 BA 82 84 59 CD E9 95 6B 2E B6 10 34 F7 AC 0....Y...k...4...
00000030 2E B2 AB 39 AD 36 92 34 52 3C EC FB 69 60 84 EF ...9.6.4R<..i`..
00000040 24 58 90 67 A6 3E D7 3F D6 6C 6C 61 AA 12 FC 4F $X.g.>.?..lla...0
00000050 E1 AA 97 59 E9 FF 23 52 F5 C2 E4 0E 73 DD F4 96 ...Y..#R....s...
00000060 A8 57 BC AA A0 53 1C FD 81 51 27 A2 A1 96 77 74 .W...S...Q'...wt
00000070 F8 F0 BD CD F7 58 7C 2A D6 36 F3 63 23 35 83 3C .....X|*.6.c#5.<
00000080 AE 0F 3E 8A 8B 9D D2 05 CF 7F 28 96 9B 24 7B 02 ..>.....(..${.

```

Figura 11.24: input1 camellia-128 modo CBC contraseña opción -nosalt

### 11.3.3. Cifrado Camelia-128 modo OFB con contraseña opción -nosalt

Comando utilizado para cifrar con Camelia-128 modo OFB con contraseña opción -nosalt:

```

patri@patri:~/Escritorio$ openssl camellia-128-ofb -pass pass:123456 -in
input1.bin -out o1_camelia_ofb_nosalt.bin -nosalt

```

Figura 11.25: Comando Camelia-128 modo OFB con contraseña



00000000	AC 86 26 6A 9C FF AC EC C4 90 6A 49 E3 C0 87 80	.&j.....jI....
00000010	39 08 DC 45 FA 5A CD 2A 79 8B 5C CC EF A0 A9 EA	9..E.Z.*y.\.....
00000020	86 5C 19 C1 A6 42 8B 14 88 EB 41 1B 29 8B 4C 96	.\...B....A.)..L.
00000030	FC 64 4A 38 63 33 57 67 90 BF 7A 07 49 48 EA 33	.dJ8c3Wg...z.IH.3
00000040	8C 07 50 96 1E 14 E3 F8 E2 A7 55 9F E5 16 4E E9	..P.....U...N.
00000050	8C 5A BE E9 60 1B 37 7C B0 62 C2 A8 DA 4A F9 6B	.Z...`.7 .b...J.k
00000060	FA 21 60 56 87 6B 66 9F 14 CA 1D 6A AD 12 8C 55	.!`V.kf....j...U
00000070	42 A8 B2 73 71 7C 89 43 15 77 EC AE D1 C8 05 F9	B...sq .C.w.....

Figura 11.26: input camellia-128 modo OFB con contraseña opción -nosalt

00000000	AC 86 26 6A 9C FF AC EC C4 90 6A 49 E3 C0 87 80	.&j.....jI....
00000010	39 18 DC 45 FA 5A CD 2A 79 8B 5C CC EF A0 A9 EA	9..E.Z.*y.\.....
00000020	86 5C 19 C1 A6 42 8B 14 88 EB 41 1B 29 8B 4C 96	.\...B....A.)..L.
00000030	FC 64 4A 38 63 33 57 67 90 BF 7A 07 49 48 EA 33	.dJ8c3Wg...z.IH.3
00000040	8C 07 50 96 1E 14 E3 F8 E2 A7 55 9F E5 16 4E E9	..P.....U...N.
00000050	8C 5A BE E9 60 1B 37 7C B0 62 C2 A8 DA 4A F9 6B	.Z...`.7 .b...J.k
00000060	FA 21 60 56 87 6B 66 9F 14 CA 1D 6A AD 12 8C 55	.!`V.kf....j...U
00000070	42 A8 B2 73 71 7C 89 43 15 77 EC AE D1 C8 05 F9	B...sq .C.w.....

Figura 11.27: input1 camellia-128 modo OFB contraseña opción -nosalt

Al igual pasa en estos tres modos como en aes en el ejercicio 6, hemos añadido la opción -nosalt, por lo que ya no tendremos la cadena aleatoria que se añade a la contraseña para derivar la clave y estamos expuestos a ataques de diccionario. Tenemos un bloque menos, en los tres modos con respecto al punto anterior en el que por defecto añade la opción -salt. Y como bien hemos visto anteriormente, ECB y CBC con padding y OFB sin la necesidad y un bloque menos aún que estos dos anteriores.