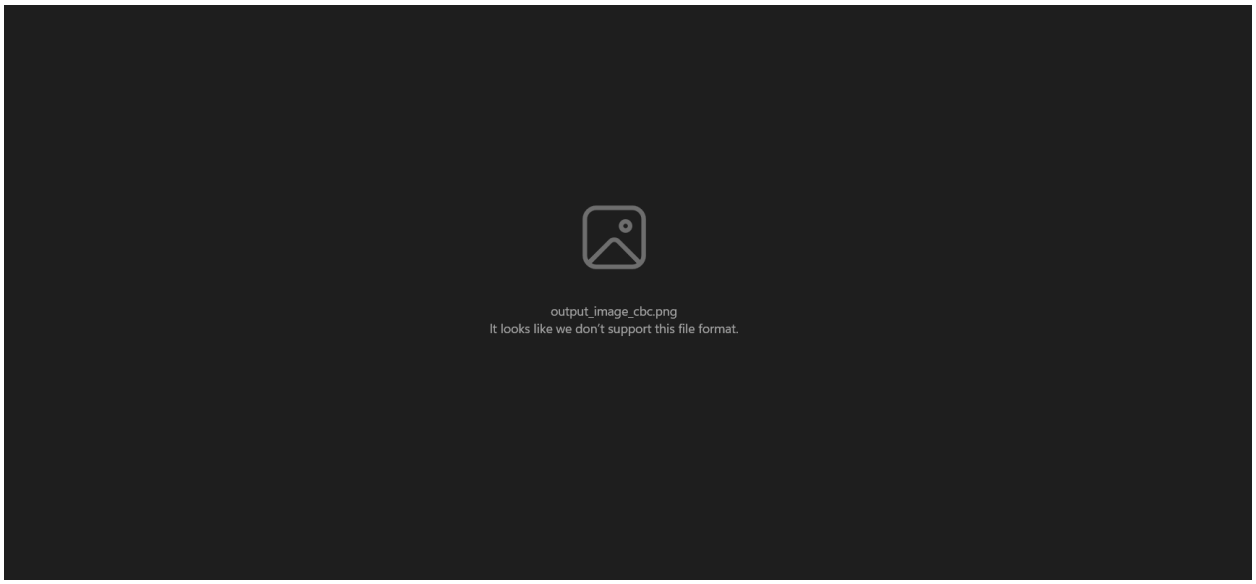
Universidad del Valle de Guatemala	(CC3078) Cifrado de Información
Santiago Taracena Puga (20017)	Laboratorio 3 - Cifrados Simétricos

Parte 1



El cifrado ECB (Electronic Codebook) y el cifrado CBC (Cipher Block Chaining) son dos modos de operación comunes en el cifrado simétrico, como AES (Advanced Encryption Standard). Ambos modos tienen sus propias características y ventajas, pero también presentan diferencias significativas en términos de seguridad y aplicaciones.

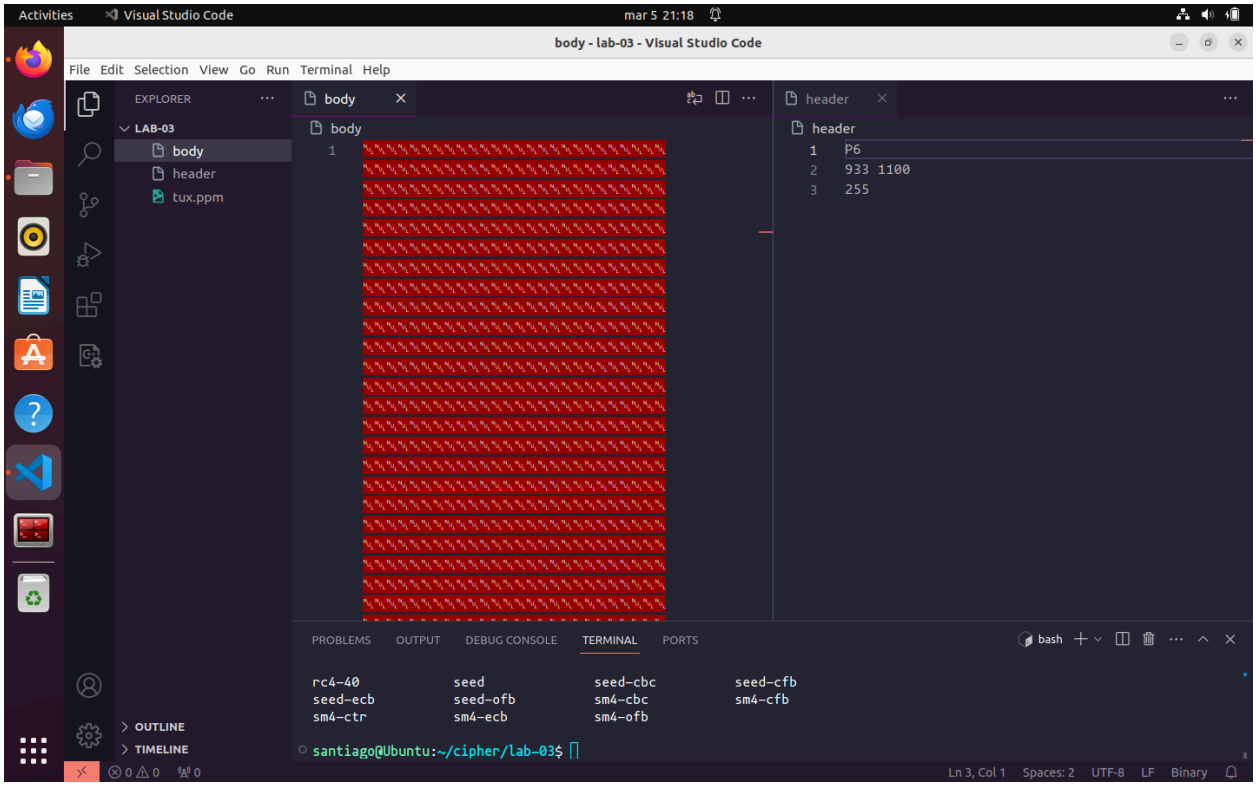
En el modo ECB, cada bloque de datos se cifra de forma independiente utilizando la misma clave. Esto significa que si un mismo bloque de datos se repite en diferentes partes del mensaje original, el cifrado resultante será idéntico para esos bloques. Esta característica puede ser explotada por un atacante para obtener información sobre el mensaje original, incluso sin tener acceso a la clave de cifrado. Además, el ECB no proporciona ninguna protección contra la reordenación de bloques de datos. Por lo tanto, no es recomendable para cifrar grandes cantidades de datos o para aplicaciones donde la seguridad es una preocupación primordial.

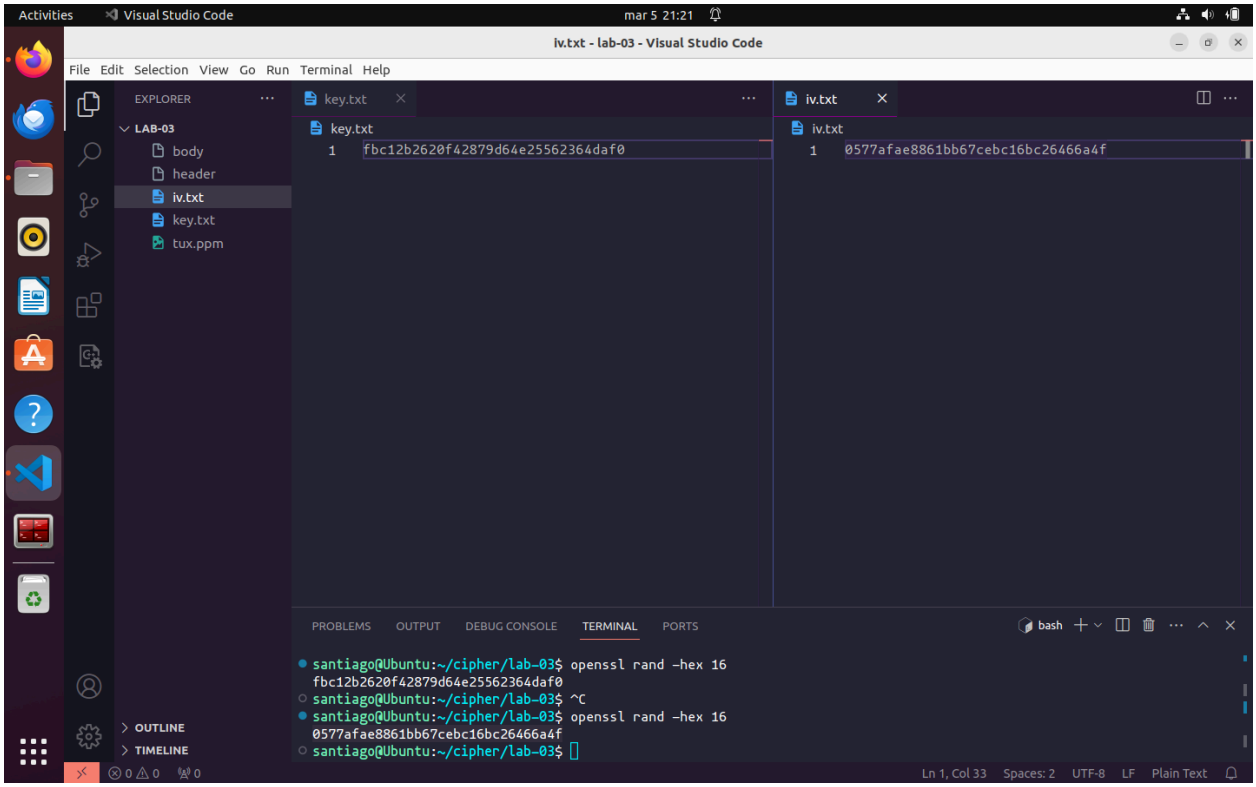
Por otro lado, el modo CBC utiliza un vector de inicialización (IV) único para cada mensaje y un esquema de retroalimentación donde el cifrado de un bloque depende del resultado del cifrado del bloque anterior. Esto hace que cada bloque de datos esté enlazado al anterior, lo que proporciona una mayor seguridad en comparación con ECB. El uso del IV también asegura que incluso si los mismos bloques de datos se encuentran en diferentes partes del mensaje original, el cifrado resultante será diferente. Sin embargo, es fundamental que el IV sea aleatorio y no predecible para garantizar la seguridad del cifrado. En caso de que el IV no esté disponible para el proceso de descifrado, como en el escenario que enfrentamos donde no había un vector inicial para CBC, no se puede realizar la operación de descifrado con éxito.

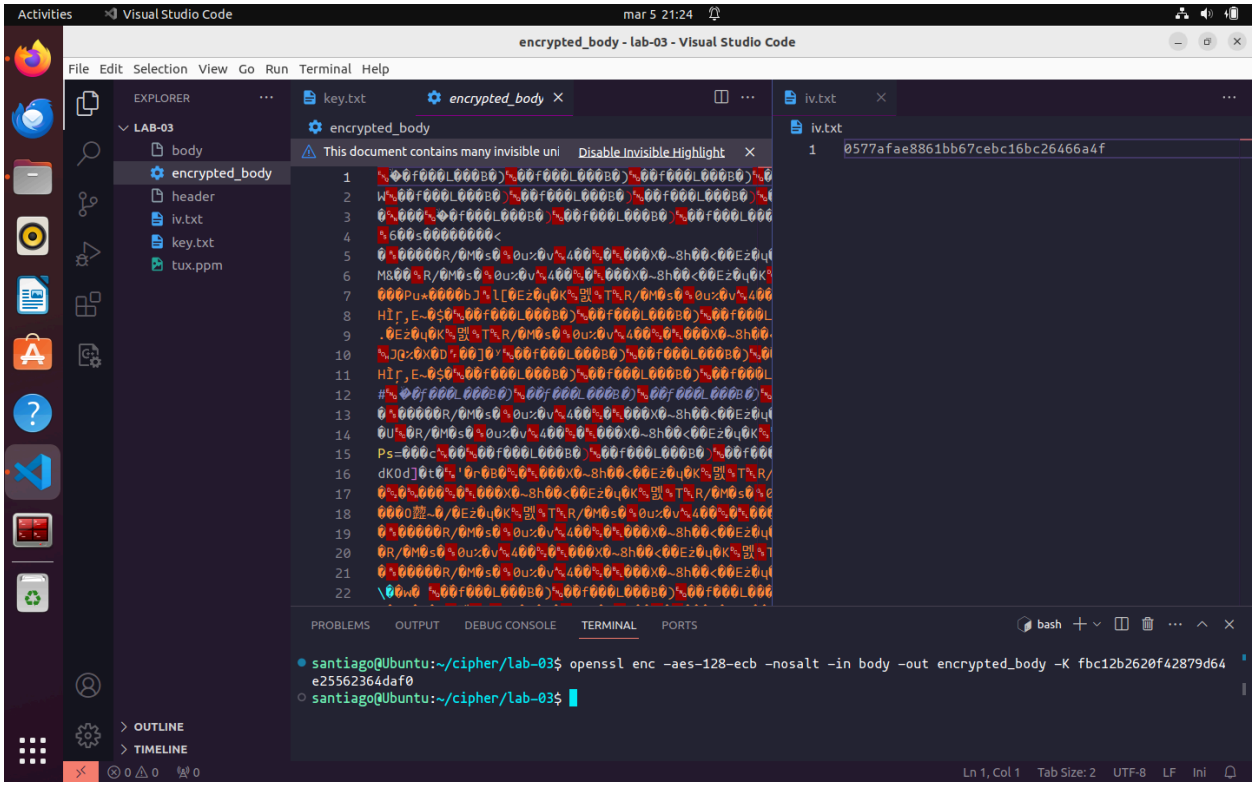
En conclusión, mientras que ECB es más simple y puede ser adecuado para ciertas aplicaciones donde la seguridad no es una preocupación crítica y los patrones de datos no se repiten, CBC ofrece una mayor seguridad al encadenar los bloques de datos y utilizar un IV único. Sin embargo, la falta del IV puede dificultar o incluso imposibilitar el proceso de descifrado. Es importante considerar cuidadosamente las necesidades de seguridad y los requisitos de la aplicación al elegir entre estos modos de cifrado.

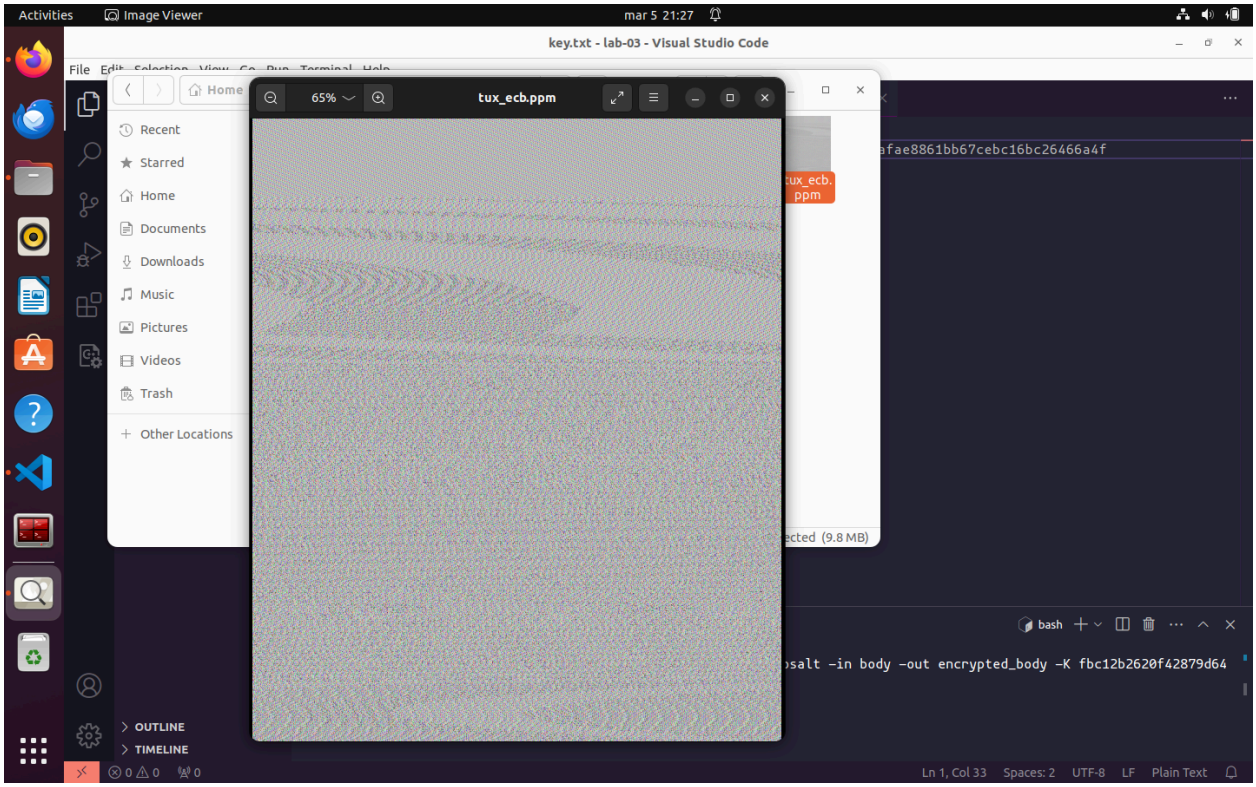
Parte 2

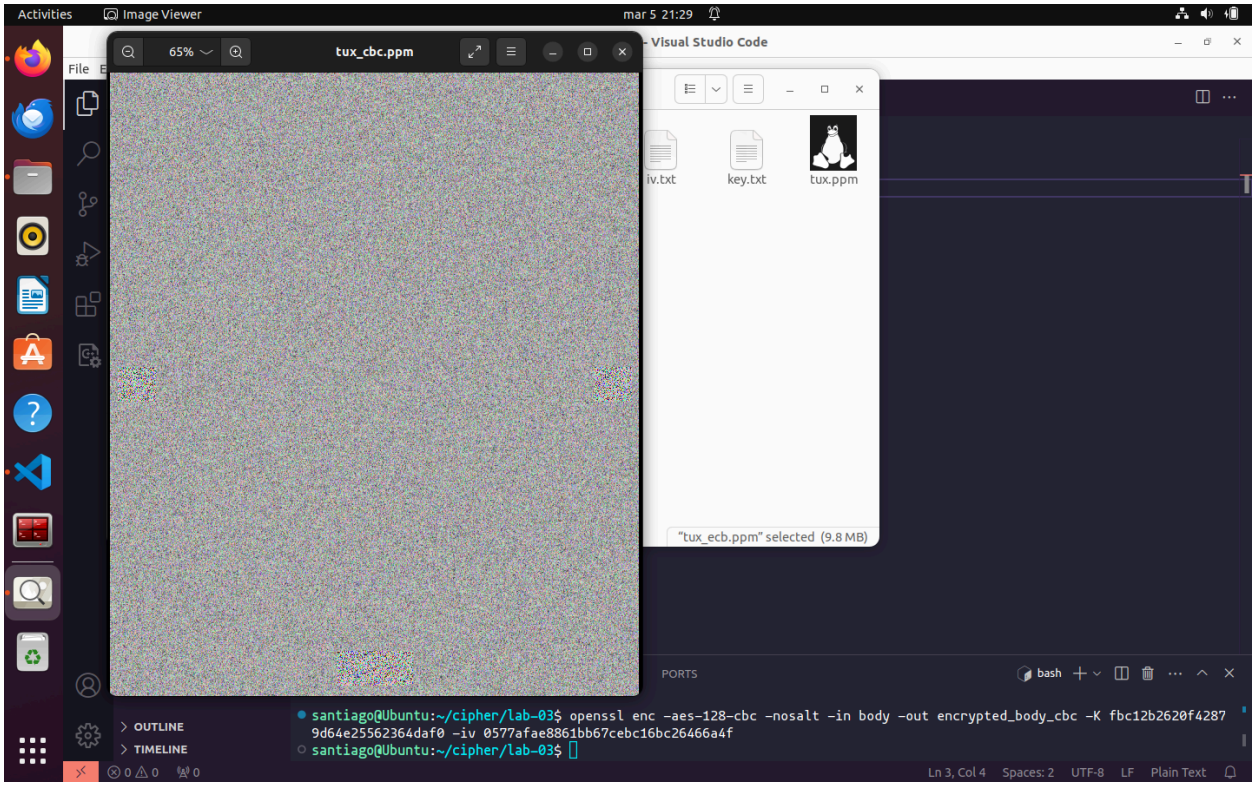
Se utilizó la librería OpenSSL en conjunto con Ubuntu con el objetivo de llevar a cabo las instrucciones ubicadas en la parte 2 del laboratorio. A continuación se encuentran capturas de pantalla con todo el procedimiento llevado a cabo y los resultados obtenidos al finalizar con la parte 2.


Activities
Visual Studio Code
mar 5 21:18
body - lab-03 - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
LAB-03
body
header
tux.ppm
body
header
1 P6
2 933 1100
3 255
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
bash
rc4-40 seed seed-cbc seed-cfb
seed-ecb seed-ofb sm4-cbc sm4-cfb
sm4-ctr sm4-ecb sm4-ofb
santiago@Ubuntu:~/cipher/lab-03$
OUTLINE
TIMELINE
Ln 3, Col 1 Spaces: 2 UTF-8 LF Binary


Activities
Visual Studio Code
mar 5 21:21
iv.txt - lab-03 - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
LAB-03
body
header
iv.txt
key.txt
tux.ppm
key.txt
1 fbc12b2620f42879d64e25562364daf0
iv.txt
1 0577afae8861bb67cebc16bc26466a4f
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
bash
santiago@Ubuntu:~/cipher/lab-03$ openssl rand -hex 16
fbc12b2620f42879d64e25562364daf0
santiago@Ubuntu:~/cipher/lab-03$ ^C
santiago@Ubuntu:~/cipher/lab-03$ openssl rand -hex 16
0577afae8861bb67cebc16bc26466a4f
santiago@Ubuntu:~/cipher/lab-03$
OUTLINE
TIMELINE
Ln 1, Col 33 Spaces: 2 UTF-8 LF Plain Text


Activities
Visual Studio Code
mar 5 21:24
encrypted_body - lab-03 - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
LAB-03
body
encrypted_body
header
iv.txt
key.txt
tux.ppm
key.txt
encrypted_body
This document contains many invisible uni
Disable Invisible Highlight
iv.txt
0577afae8861bb67cebc16bc26466a4f
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
bash
santiago@Ubuntu:~/cipher/lab-03$ openssl enc -aes-128-ecb -nosalt -in body -out encrypted_body -K fbc12b2620f42879d64
e25562364daf0
santiago@Ubuntu:~/cipher/lab-03$
OUTLINE
TIMELINE
Ln 1, Col 1
Tab Size: 2
UTF-8
LF
Ini


Activities
Image Viewer
mar 5 21:27
key.txt - lab-03 - Visual Studio Code
tux_ecb.ppm
65%
Recent
Starred
Home
Documents
Downloads
Music
Pictures
Videos
Trash
Other Locations
afae8861bb67cebc16bc26466a4f
bash
salt -in body -out encrypted_body -K fbc12b2620f42879d64
OUTLINE
TIMELINE
Ln 1, Col 33
Spaces: 2
UTF-8
LF
Plain Text


Activities
Image Viewer
mar 5 21:29
Visual Studio Code
65%
tux_cbc.ppm
iv.txt
key.txt
tux.ppm
"tux_ecb.ppm" selected (9.8 MB)
PORTS
bash
OUTLINE
TIMELINE
santiago@Ubuntu:~/cipher/lab-03$ openssl enc -aes-128-cbc -nosalt -in body -out encrypted_body_cbc -K fbc12b2620f4287
9d64e25562364daf0 -iv 0577afae8861bb67cebc16bc26466a4f
santiago@Ubuntu:~/cipher/lab-03$
Ln 3, Col 4
Spaces: 2
UTF-8
LF
Plain Text