

Lab 3 - Task-1: AES Encryption Using Different Modes (OpenSSL)

Objective

Encrypt the same plaintext with AES-128 using three modes (ECB, CBC, CFB). Then decrypt and verify that the plaintext is recovered. Include the exact commands.

Steps Performed

- Created a plaintext file from the terminal (no gedit required).
- Defined a 128-bit key and a 128-bit IV in hexadecimal.
- Encrypted the file using OpenSSL enc with three AES modes (ECB, CBC, CFB).
- Decrypted and verified using diff (no output means identical).
- Optionally viewed ciphertext bytes using xxd or GHex.

Environment

- OS: Ubuntu Linux
- Tools: OpenSSL, nano/echo, xxd (or GHex)
- AES parameters: 128-bit key; 128-bit IV for CBC/CFB
- Note: Use -nosalt with explicit -K/-iv for deterministic outputs.

Create Plaintext File

Option A (nano):

```
nano plain.txt      # type lines, then Ctrl+O, Enter, Ctrl+X
```

Option B (echo):

```
echo -e "This is a simple plaintext file.\nIt will be encrypted using AES.\nTesting ECB, CBC, and CFB modes." > plain.txt
```

Preview:

```
cat plain.txt
```

Set Key and IV (HEX)

KEY=00112233445566778899aabbcdddeeff

IV=0102030405060708090a0b0c0d0e0f10

Both are 16 bytes (32 hex characters).

Encryption Commands

```
# AES-128-ECB (no IV)
openssl enc -aes-128-ecb -e -in plain.txt -out ct_ecb.bin -K $KEY -nosalt
```

```
# AES-128-CBC (needs IV)
openssl enc -aes-128-cbc -e -in plain.txt -out ct_cbc.bin -K $KEY -iv $IV -nosalt
```

```
# AES-128-CFB (needs IV)
openssl enc -aes-128-cfb -e -in plain.txt -out ct_cfb.bin -K $KEY -iv $IV -nosalt
```

Lab 3 - Task-1: AES Encryption Using Different Modes (OpenSSL)

Decryption and Verification

```
openssl enc -aes-128-ecb -d -in ct_ecb.bin -out dec_ecb.txt -K $KEY -nosalt
openssl enc -aes-128-cbc -d -in ct_cbc.bin -out dec_cbc.txt -K $KEY -iv $IV -nosalt
openssl enc -aes-128-cfb -d -in ct_cfb.bin -out dec_cfb.txt -K $KEY -iv $IV -nosalt

# Compare decrypted and original
diff plain.txt dec_ecb.txt; diff plain.txt dec_cbc.txt; diff plain.txt dec_cfb.txt
```

Observations

- All three decrypted files matched the original (diff printed nothing).
- ECB requires no IV; CBC and CFB require IV.
- Ciphertext files are binary. To preview first bytes in hex:

```
xxd -g 1 -l 64 ct_ecb.bin | head
xxd -g 1 -l 64 ct_cbc.bin | head
xxd -g 1 -l 64 ct_cfb.bin | head
```

Submission Checklist

- 1) This PDF report.
- 2) The three ciphertext files: ct_ecb.bin, ct_cbc.bin, ct_cfb.bin.
- 3) The decrypted text files, or screenshots showing diff produced no output.
- 4) Optional: screenshots from xxd or GHex showing ciphertext in hex form.