Laboratory_03: AES Encryption using OpenSSL

This laboratory covers OpenSSL tool applicability for AES encryption with different key sizes and cipher modes.

Installation

- sudo apt update
- sudo apt-get install openssl -y
- openssl version

OpenSSL for Symmetric Encryption using AES-128-CBC

1. Create a file to encrypt:
    1. echo "lab for fun, hands-on learning." > secret.txt
2. Review the created file:
    1. cat secret.txt
3. List AES cipher algorithms:
    1. openssl list -cipher-algorithms | grep -i aes
4. Generate a random key (16 bytes = 128 bits for AES-128):
    1. openssl rand -hex 16 > aes128.key
    2. KEY=$(cat aes128.key)
5. Generate a random initialization vector (16 bytes for AES):
    1. openssl rand -hex 16 > aes.iv
    2. IV=$(cat aes.iv)
6. Encrypt the file using AES-128-CBC:
    1. openssl enc -aes-128-cbc -e -in secret.txt -out secret_aes128.enc -K $KEY -iv $IV

7. Display the encrypted content:
    1. xxd -p secret_aes128.enc
8. Decrypt file using AES-128-CBC:
    1. openssl enc -aes-128-cbc -d -in secret_aes128.enc -out secret_aes128.dec -K $KEY -iv $IV
9. Verify decryption was successful:
    1. cat secret_aes128.dec

OpenSSL for Symmetric Encryption using AES-192-CBC

1. Generate a random key (24 bytes = 192 bits for AES-192):
    1. openssl rand -hex 24 > aes192.key
    2. KEY=$(cat aes192.key)
2. Encrypt the file using AES-192-CBC:
    1. openssl enc -aes-192-cbc -e -in secret.txt -out secret_aes192.enc -K $KEY -iv $IV
3. Display the encrypted content:
    1. xxd -p secret_aes192.enc
4. Decrypt file using AES-192-CBC:
    1. openssl enc -aes-192-cbc -d -in secret_aes192.enc -out secret_aes192.dec -K $KEY -iv $IV
5. Verify decryption was successful:
    1. cat secret_aes192.dec

OpenSSL for Symmetric Encryption using AES-256-CBC

1. Generate a random key (32 bytes = 256 bits for AES-256):
    1. openssl rand -hex 32 > aes256.key
    2. KEY=$(cat aes256.key)

2. Encrypt the file using AES-256-CBC:
    1. openssl enc -aes-256-cbc -e -in secret.txt -out secret_aes256.enc -K $KEY -iv $IV
3. Display the encrypted content:
    1. xxd -p secret_aes256.enc
4. Decrypt file using AES-256-CBC:
    1. openssl enc -aes-256-cbc -d -in secret_aes256.enc -out secret_aes256.dec -K $KEY -iv $IV
5. Verify decryption was successful:
    1. cat secret_aes256.dec

Exploring Different AES Cipher Modes

AES-256-ECB (Electronic Codebook) Mode

1. Encrypt the file using AES-256-ECB (note: no IV needed for ECB):
    1. openssl enc -aes-256-ecb -e -in secret.txt -out secret_aes256_ecb.enc -K $KEY
2. Display the encrypted content:
    1. xxd -p secret_aes256_ecb.enc
3. Decrypt file using AES-256-ECB:
    1. openssl enc -aes-256-ecb -d -in secret_aes256_ecb.enc -out secret_aes256_ecb.dec -K $KEY
4. Verify decryption was successful:
    1. cat secret_aes256_ecb.dec

AES-256-CFB (Cipher Feedback) Mode

1. Encrypt the file using AES-256-CFB:
    1. openssl enc -aes-256-cfb -e -in secret.txt -out secret_aes256_cfb.enc -K $KEY -iv $IV

2. Display the encrypted content:
    1. xxd -p secret_aes256_cfb.enc
3. Decrypt file using AES-256-CFB:
    1. openssl enc -aes-256-cfb -d -in secret_aes256_cfb.enc -out secret_aes256_cfb.dec -K $KEY -iv $IV
4. Verify decryption was successful:
    1. cat secret_aes256_cfb.dec

AES-256-OFB (Output Feedback) Mode

1. Encrypt the file using AES-256-OFB:
    1. openssl enc -aes-256-ofb -e -in secret.txt -out secret_aes256_ofb.enc -K $KEY -iv $IV
2. Display the encrypted content:
    1. xxd -p secret_aes256_ofb.enc
3. Decrypt file using AES-256-OFB:
    1. openssl enc -aes-256-ofb -d -in secret_aes256_ofb.enc -out secret_aes256_ofb.dec -K $KEY -iv $IV
4. Verify decryption was successful:
    1. cat secret_aes256_ofb.dec

AES-256-CTR (Counter) Mode

1. Encrypt the file using AES-256-CTR:
    1. openssl enc -aes-256-ctr -e -in secret.txt -out secret_aes256_ctr.enc -K $KEY -iv $IV
2. Display the encrypted content:
    1. xxd -p secret_aes256_ctr.enc
3. Decrypt file using AES-256-CTR:

1. openssl enc -aes-256-ctr -d -in secret_aes256_ctr.enc -out secret_aes256_ctr.dec -K $KEY -iv $IV

4. Verify decryption was successful:
    1. cat secret_aes256_ctr.dec

Comparing Encryption Size and Performance

1. Create a larger test file:
    1. dd if=/dev/urandom of=large_file.bin bs=1M count=10
2. Measure encryption time for different algorithms:
    1. DES
        1. time openssl enc -des-cbc -e -in large_file.bin -out large_file_des.enc -K $DES_KEY -iv $DES_IV
    2. 3DES
        1. time openssl enc -des-ede3-cbc -e -in large_file.bin -out large_file_3des.enc -K $DES3_KEY -iv $DES3_IV
    3. AES-128
        1. time openssl enc -aes-128-cbc -e -in large_file.bin -out large_file_aes128.enc -K $AES128_KEY -iv $AES_IV
    4. AES-256
        1. time openssl enc -aes-256-cbc -e -in large_file.bin -out large_file_aes256.enc -K $AES256_KEY -iv $AES_IV
    5. Compare file sizes:
        1. ls -lh large_file*