Laboratory_05: OpenSSL for AES, RSA and ECC

This laboratory covers OpenSSL tool applicability.

Installation

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

OpenSSL for symmetric encryption

1. Create a file to encrypt:
    a. echo "lab for fun, hands-on learning." > secret.txt
2. Review the created file:
    a. cat secret.txt
3. Encrypt the file using AES:
    a. openssl enc -aes-256-cbc -salt -in secret.txt -out secret.enc -pbkdf2
        1. openssl enc lets OpenSSL know that we want to encrypt something.
        2. -aes-256-cbc is the encryption method we are using.
        3. -salt adds some random data to make our encryption even more secure.
        4. -in secret.txt is the file we want to encrypt.
        5. -out secret.enc is the name of the encrypted file we will create.
        6. -pbkdf2 uses a special method to create a strong key from our password.
4. Review encrypted file:
    a. ls -l secret.enc
    b. cat secret.enc

5. Decrypt the file:
    a. openssl enc -aes-256-cbc -d -in secret.enc -out decrypted.txt -pbkdf2
6. Review the file:
    a. cat decrypted.txt
7. Ensure decrypted message:
    a. diff secret.txt decrypted.txt

OpenSSL for RSA

8. Generate a key pair with:
    a. openssl genrsa -out private.pem 1024
9. Review key pair created:
    a. cat private.pem
10. View the RSA key pair:
    a. openssl rsa -in private.pem -text
11. Secure the encrypted key with 3-DES:
    a. openssl rsa -in private.pem -des3 -out key3des.pem
12. Export the public key:
    a. openssl rsa -in private.pem -out public.pem -outform PEM –pubout
13. Create a file named “myfile.txt” and put a message into it. Next encrypt it with your public key:
    a. openssl rsautl -encrypt -inkey public.pem -pubin -in myfile.txt -out file.bin
14. n decrypt with the private key:
    a. openssl rsautl -decrypt -inkey private.pem -in file.bin -out decrypted.txt

OpenSSL for Elliptic Curve Cryptography (ECC)

15. Generate a private key:

a. openssl ecparam -name secp256k1 -genkey -out priv.pem

16. View the details of the ECC parameters used with:

a. openssl ecparam -in priv.pem -text - param_enc explicit –noout

17. Generate your public key based on your private key with:

a. openssl ec -in priv.pem -text -noout