CT437 – Worksheet Week 8

Block Ciphers and Public Key Cryptography using OpenSSL

Overview:

In this tutorial you will study the OpenSSL API for various encryption tasks. Please make sure that you have a Linux VM up and running with gcc and OpenSSL installed (this is the default for most Linux distributions).

Problem 1: Getting started / Block ciphers in OpenSSL

1. Check the gcc compiler via ***gcc -v***
2. Check out your openssl version via ***openssl version***
3. Install openssl libs via ***sudo apt-get install libssl-dev***
4. Compile and execute the two source code files, following the instructions in the file header. ***gcc <source file> -o <destination file> -lcrypto***
5. Review the code.

Problem 2: RSA encryption

1. Execute and subsequently review the following command:
   openssl genrsa -aes128 -out <your name>_keyfile.pem 1024
   In detail, determine
   a. the kind of key pair generated
   b. the nature of a .pem file (check out Wikipedia)
   c. what aes128 has to do with all this
2. Check out the generated pem file content via
   head <your name>_keyfile.pem
   The key is encoded via BASE64. Explain this data format.
3. Extract the generated public key via
   openssl rsa -in <your name>_keyfile.pem -pubout > public_key.txt
   and save it in a file.
4. Exchange your public key with your classmates (e.g., via email)
5. Create a secret message and encode it using
   openssl pkeyutl -encrypt -inkey <receiver's public key> -pubin -in <secret file> -out <secret file>.enc
6. View the generated ciphertext file via ***hexdump -C <secret file>***
7. Exchange the ciphertext file.
8. Decode your received encrypted files (using your private key) via
   openssl pkeyutl -decrypt -inkey <your name>_keyfile.pem -in <secret file> -out <output file>

Problem 3: ECC encryption with ECDH

1. Generate an ECC key pair via
   openssl ecparam -name secp256k1 -genkey -noout -out <your name>_ecckey.pem
2. View the key pair via ***openssl ec -noout -text -inform PEM -in <your name>_ecckey.pem***
3. Extract the public key via
   openssl ec -in <your name>_ecckey.pem -pubout -out <your name>_ecckeypublic.pem
4. Share this public key as done in problem 2.
5. Generate a ECDH key via
   openssl pkeyutl -derive -inkey <your name>_ecckey.pem -peerkey <received>_ecckeypublic.pem -out shared_secret.bin
6. Compare the generated key with your peers via ***base64 shared_secret.bin***