Lab 4: Programming Symmetric & Asymmetric Crypto

Objective

The objective of this lab is to implement various cryptographic operations programmatically to understand how symmetric and asymmetric encryption, hashing, and digital signatures work internally. The lab also includes performance analysis based on key length and execution time.

References

https://mojoauth.com/encryption-decryption/aes-256-encryption--python/

https://stackoverflow.com/questions/30056762/rsa-encryption-and-decryption-in-python

https://mojoauth.com/hashing/sha-256-in-python/

Tools and Environment

- **Programming Language:** Python 3
- **Libraries Used:**
    - `pycryptodome` for AES and RSA operations
    - `hashlib` for SHA-256 hashing
    - `time` for measuring execution time
- **System:** macOS M1 / Ubuntu Linux
- **Editor:** Visual Studio Code

Tasks and Implementation

Task 1: AES Encryption and Decryption (5 marks)

- Implemented AES encryption and decryption using:
    - Key lengths: **128-bit** and **256-bit**
    - Modes: **ECB** and **CFB**
- The program encrypts a file, stores the ciphertext in another file, and then decrypts it to display the plaintext on the console.

Key Features:

- AES key is generated once and saved in `aes_key.bin`.
- Uses PKCS#7 padding for block alignment.
- Execution time is recorded in seconds and displayed along with bits processed per second.

Task 2: RSA Encryption and Decryption (4 marks)

- RSA key pair is generated once and saved as:
- The user selects files for encryption and decryption through the command line.
- Encrypted data is written to a file; decrypted data is displayed on the console.

Task 3: RSA Signature (4 marks)

- The program generates a digital signature using RSA private key and verifies it using the corresponding public key.
- The signature is saved in a separate `.sig` file.

Task 4: SHA-256 Hashing (3 marks)

- The program computes the SHA-256 hash of user input or a given file.
- Hashing is done using Python's built-in `hashlib` library.