Lab 6: Secure Programming

Write a program that integrates multiple cryptographic functionalities into a single application. The program should allow users to perform the following operation:

1. **AES Encryption/Decryption:** Should support AES encryption and decryption using either ECB or CBC mode. Users should be able to input the plaintext or ciphertext, specify the AES mode, provide the necessary keys, and select the operation (encryption or decryption). The program should deliver the expected output based on the user's input.

   Required inputs: Mode (ECB/CBC), Operation (Encrypt/Decrypt), plaintext/ciphertext, Key

   https://pycryptodome.readthedocs.io/en/latest/src/cipher/aes.html

2. **RSA Encryption/Decryption:** Should offer RSA encryption and decryption capabilities. Users should be able to input the plaintext or ciphertext, select the RSA operation, provide the required RSA keys, and choose the desired encryption or decryption process.

   Required inputs: Operation (Encrypt/Decrypt), Plaintext/ciphertext, Key

   https://www.section.io/engineering-education/rsa-encryption-and-decryption-in-python/

3. **Hashing:** Should include hashing functionalities supporting SHA1 and SHA256 algorithms. Users should be able to input the data to be hashed, specify the hash mode (SHA1 or SHA256), and trigger the operation. The program should generate the corresponding hash value.

   Required inputs: Plaintext, hash mode(SHA1 / SHA256)

   https://geekflare.com/secure-hashing-with-python-hashlib/
   https://docs.python.org/3/library/hashlib.html

4. **Digital Signature using RSA:** Should facilitate the generation and verification of digital signatures using RSA. Users should be able to input the message to be signed, select the RSA keys for signing, and initiate the signing process. Similarly, users should be able to input the signed message, specify the RSA public key, and perform the digital signature verification. The program should ensure the integrity and authenticity of the digital signatures.

   Required inputs: Operation(Generation/Verification), Message / signature

https://riptutorial.com/python/example/19025/generating-rsa-signatures-using-pycrypto

5. **MAC Generation:** The program should allow users to generate Message Authentication Codes (MAC) for data integrity and authenticity. Users should be able to input the data, select the MAC algorithm, and initiate the MAC generation process. The program should generate the MAC value accurately, maintaining the security of the data.

   Required inputs: Message

https://pycryptodome.readthedocs.io/en/latest/src/hash/cmac.html