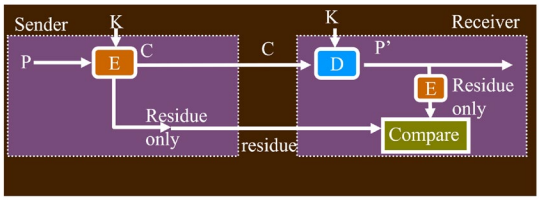
Homework 2 (100 pts)

Rule: Finish all of the following on your own. Submit your solution in PDF format.

1. What are the properties of hash functions?
2. Explain meet-in-the-middle attacks against double-DES.
3. Explain why the following attempt cannot ensure integrity.



4. Alice designs a new double-DES scheme. The scheme will first DES-encrypt a message using K1 to get an intermediate ciphertext, then DES-**decrypt** the intermediate ciphertext using K2 to get the final ciphertext. Is there any vulnerability in Alice's design?
5. Suppose the sub-key generation function is to reverse all the bits of the key K (e.g., 0 1 1 1 ➔ 1 1 1 0), and the scrambling function is f = M XOR K', where M is the second half of input bits and K' is the sub-key (i.e., the reverse of the original key K).
   Now given the original K = 0011, and input bits 1111 0000, compute the output of the Feistel Cipher.
6. A and B want to ensure the integrity and authenticity of the messages between them, but they do NOT care about the confidentiality. Assume A and B share a key K. Answering two questions
   a. How can they achieve their goal only with symmetric key cryptography?
   b. How can they achieve their goal with hash function H?
7. Bob is assigned a task to design a way to allow encryption of files stored in the system: all files are stored in an encrypted form. If a block of a file is requested, the system should retrieve the block, decrypt it and return the plaintext to the host. Similarly, if a host writes a block to the storage system, the system should retrieve the right keying material, encrypt the block, and only save the ciphertext on disk. Consider the modes of operations discussed in class (i.e., ECB, CBC, CFB, CTR). Which one should be used in terms of read/write efficiency? Why? (You do NOT need to consider the key storage problem.)
8. **Lab Task: Exploring AES-CBC Encryption Using OpenSSL**
   In this lab, you will explore encryption using OpenSSL. First, ensure you have OpenSSL installed. If you are using Linux or macOS, OpenSSL is usually pre-installed; you can check by running **openssl version**. If it is not installed, use **sudo apt install openssl** (Ubuntu) or **brew install openssl** (macOS). On Windows, download OpenSSL from https://slproweb.com/products/Win32OpenSSL.html (Or using WSL on Windows) and follow the installation instructions. Once installed, verify by running **openssl** version in the terminal or command prompt.

Tasks:

1. Basic Encryption and Decryption
   - Use OpenSSL to encrypt a text file (message.txt) with AES-256-CBC encryption using a password.
   - Decrypt the file and verify the content.
   - Provide a screenshot of the commands and output.
2. Encrypt and Decrypt Using a Key File
   - Generate a random encryption key using OpenSSL and save it to a file.
   - Use this key to encrypt message.txt and decrypt it back.
   - Show the encryption key, encrypted file, and decrypted file in a screenshot.
3. Observing Encrypted Data
   - Encrypt message.txt using AES-256-CBC and then open the encrypted file in a text editor.
   - Compare the original and encrypted content, then describe your observations.