README

File Encryption and Decryption using AES in C++
This C++ code demonstrates how to encrypt and decrypt a file using the Advanced Encryption Standard (AES) algorithm, a widely used symmetric encryption algorithm.

How it Works

1. **Include Necessary Libraries**:
    - The code includes the Crypto++ library which is use for the AES encryption and file handling.
2. **Key and Initialization Vector (IV)**:
    - AES encryption requires a key and an Initialization Vector (IV).
    - In this code, both the key and IV are specified as arrays of 16 bytes or 128bits. It's important to keep these values secure.
3. **EncryptFile Function**:
    - The **EncryptFile** function takes input file path, output file path, key, and IV as parameters.
    - It initializes AES encryption with the provided key and IV.
    - The function then opens the input file for reading in binary mode (**ios::binary**) and the output file for writing in binary mode.
    - It creates an **AES CBC (Cipher Block Chaining)** encryption object by referring to the Crypto++ library and then associates it with the input file and output file.
    - **Padding** (a method of filling the data to match the block size) is added to ensure the data length is a multiple of the block size.
    - It reads the input file into a buffer, encrypts the data, and writes the encrypted data to the output file.
4. **DecryptFile Function**:
    - The **DecryptFile** function is similar to **EncryptFile**, but it performs decryption.
    - It initializes AES decryption with the provided key and IV.
    - Padding is removed after decryption to recover the original data length.
5. **Main Function**:
    - In the **main** function, a sample file, key, Initialization Vector, input file, encrypted file, and decrypted file paths are provided.
    - It encrypts by referring to the **EcryptFile Function** the input file and outputs a message indicating success.
    - Then, it decrypts the encrypted file by referring to **DecryptFile Function** and outputs a success message.