**### KeyManager Class**

- \*\*`read\_key(key\_file: str) -> bytes`\*\*: Reads a key from a file and returns it as bytes.

- \*\*`save\_key(key\_file: str, key: bytes)`\*\*: Saves a key to a file.

- \*\*`generate\_key(key\_len=256) -> bytes`\*\*: Generates a random key of the specified length in bits and returns it as bytes.

**### Utility Functions**

- \*\*`bitize(byts: bytes) -> list[int]`\*\*: Converts bytes into a list of bits.

- \*\*`debitize(bits: Iterable[int]) -> bytes`\*\*: Converts a list of bits into bytes.

- \*\*`bit2hex(bits: Iterable[int]) -> str`\*\*: Converts bits into a hexadecimal string.

- \*\*`hex2bit(hex\_str: str) -> list[int]`\*\*: Converts a hexadecimal string into bits.

**### DES Class**

- \*\*`key\_generation(key: list[int]) -> list[list[int]]`\*\*: Generates 16 round subkeys from the initial 64-bit key.

#### Encryption Functions

- \*\*`enc\_block(block: list[int]) -> list[int]`\*\*: Encrypts a 64-bit block using DES.

- \*\*`mixer(L: list[int], R: list[int], sub\_key: list[int]) -> list[int]`\*\*: Mixes the left and right halves of the block.

- \*\*`swapper(L: list[int], R: list[int]) -> tuple[list[int]]`\*\*: Swaps the left and right halves of the block.

#### Decryption Functions

- \*\*`dec\_block(block: list[int]) -> list[int]`\*\*: Decrypts a 64-bit block using DES.

#### Main Encryption and Decryption Methods

- \*\*`encrypt(msg\_str: str) -> bytes`\*\*: Encrypts the entire message, padding if necessary.

- \*\*`decrypt(msg\_bytes: bytes) -> str`\*\*: Decrypts the entire message, removing padding if necessary.

**### Main Program Flow**

1. \*\*Initialization\*\*:

- Create an instance of the `KeyManager` to read or generate keys.

- Instantiate the `DES` class with the key.

2. \*\*Key Generation\*\*: Generate the round subkeys using the `key\_generation` method.

3. \*\*Encryption\*\* (Server):

- Receive a message from the client.

- Encrypt the message using `encrypt` method.

- Send the ciphertext back to the client.

4. \*\*Decryption\*\* (Client):

- Receive the ciphertext from the server.

- Decrypt the ciphertext using the `decrypt` method.

- Display the decrypted message.

**### Error Handling**

- Properly handle exceptions such as key length, block size, and input validation.

**### What I have Learned**

Through this lab, I have learned the fundamental principles of data encryption using the Data Encryption Standard (DES) algorithm, including key generation, message padding, and the step-by-step encryption and decryption processes. And I also learned how to build a connection using socket for client and server, and using some basic connection method such as recv and send.

**###** **Testing result**  
电脑萤幕的截图

描述已自动生成