Course	COMP 7005
Program	Bachelor of Science in Applied Computer Science
Term	January 2025

- This is an individual programming assignment.

Objective

- Develop a client/server application using network sockets for inter-process communication.
- This assignment emphasizes file transfer, Vigenère cipher encryption, and socket programming concepts.

Learning Outcomes

- Understand and implement network sockets for client-server communication.
- Apply Vigenère cipher encryption to transform data.
- Gain experience in designing and testing inter-process communication programs.

Assignment Details

Requirements

You will write two programs.

Client

- Accepts the following inputs as command-line arguments:
    - The IP address and port number of the server.
    - The name of the file to send to the server.
    - The keyword for the Vigenère cipher.
- Reads the contents of the specified file.
- It connects to the server via a network socket using the given IP address and port number.
- Sends the keyword and the file's content to the server.
- Receives the encrypted file content from the server and prints it to the terminal.

Server

- Accepts the following inputs as command-line arguments:
    - The IP address and port number to bind the server.
- Listens for client connections on the specified IP and port using a network socket.
- Receives the keyword and file content from the client.
- Encrypts the file content using the Vigenère cipher with the received keyword.
- Sends the encrypted content back to the client.

Constraints

- You may use any language you like.
- The program must run on a UNIX-like Operating System (e.g., Linux or macOS).
- Use network sockets exclusively for client-server communication.
- Ensure proper error handling for socket operations.
- Implement cleanup mechanisms to handle closed or failed connections.
- Implement the Vigenère cipher encryption manually.
- Assign numerical values to letters (A = 0, B = 1, ..., Z = 25).
- Use modulo 26 arithmetic for wraparound behaviour.
- Encrypt only alphabetic characters, preserving non-alphabetic characters as-is.
- Do not use higher-level libraries for socket programming.
- Do not use external encryption libraries; implement the Vigenère cipher algorithm from scratch.

Resources

- Man Pages: man 2 socket, man 3 read, man 3 write, man 3 getopt.
- Vigenère Cipher Explanation: Wikipedia.
- Code samples from your course materials.

Submission

- Ensure your submission meets all the guidelines, including formatting, file type, and submission.
- Follow the AI usage guidelines.
- Be aware of the late submission policy to avoid losing marks.
- ***Note: Please strictly adhere to the submission requirements to ensure you don't lose any marks.***

Evaluation

Topic	Value

Correct implementation of the client program	15%
Correct implementation of the server program	15%
Proper use of Network Sockets	20%
Design	20%
Testing	30%
Total	100%

Hints

- Test your programs with small and large files to ensure correctness and reliability.
- Use temporary files for debugging to monitor data sent and received over the socket.
- Consider edge cases in encryption, such as files containing no alphabetic characters.
- Ensure proper cleanup of sockets and connections after the server shuts down to prevent errors on subsequent runs.
- Validate IP and port inputs on the client and server to avoid unexpected runtime errors.
- For testing on one machine, run the server locally and connect the client using `127.0.0.1` or the machine's local IP address.
- Ensure that you test with a multiple client machined and a server machine.