**4. Pseudocode:**

**Server:**

function main() {

createSocket();

bindSocket(PORT);

listenForConnections();

while (true) {

clientSocket = acceptConnection();

handleClient(clientSocket);

}

}

function handleClient(socket) {

if (authenticateClient(socket)) {

request = receiveRequest(socket);

switch (request.type) {

case LIST\_FILES:

sendFileList(socket);

break;

case GET\_FILE:

sendFile(socket, request.filename);

break;

case UPLOAD\_FILE:

receiveFile(socket, request.filename);

break;

}

}

closeSocket(socket);

}

function authenticateClient(socket) {

password = receiveData(socket);

return (password == PASSWORD);

}

function sendFileList(socket) {

files = listFiles();

sendData(socket, files);

}

function sendFile(socket, filename) {

data = readFile(filename);

encryptedData = encryptData(data);

sendData(socket, encryptedData);

}

function receiveFile(socket, filename) {

data = receiveData(socket);

decryptedData = decryptData(data);

writeFile(filename, decryptedData);

}

**4 Pseudocode**

**Client:**

function main() {

createSocket();

connectToServer(IP, PORT);

authenticateServer();

command = getUserCommand();

sendRequest(command);

if (command.type == LIST\_FILES) {

files = receiveFileList();

displayFiles(files);

} else if (command.type == GET\_FILE) {

data = receiveFile();

writeFile(command.filename, data);

} else if (command.type == UPLOAD\_FILE) {

data = readFile(command.filename);

sendFile(data);

}

closeSocket();

}

function authenticateServer() {

sendData(PASSWORD);

response = receiveData();

if (response != SUCCESS) {

exit();

}

}

function sendRequest(command) {

sendData(command);

}

function receiveFileList() {

return receiveData();

}:

function receiveFile() {

encryptedData = receiveData();

return decryptData(encryptedData);

}

function sendFile(data) {

encryptedData = encryptData(data);

sendData(encryptedData);

}