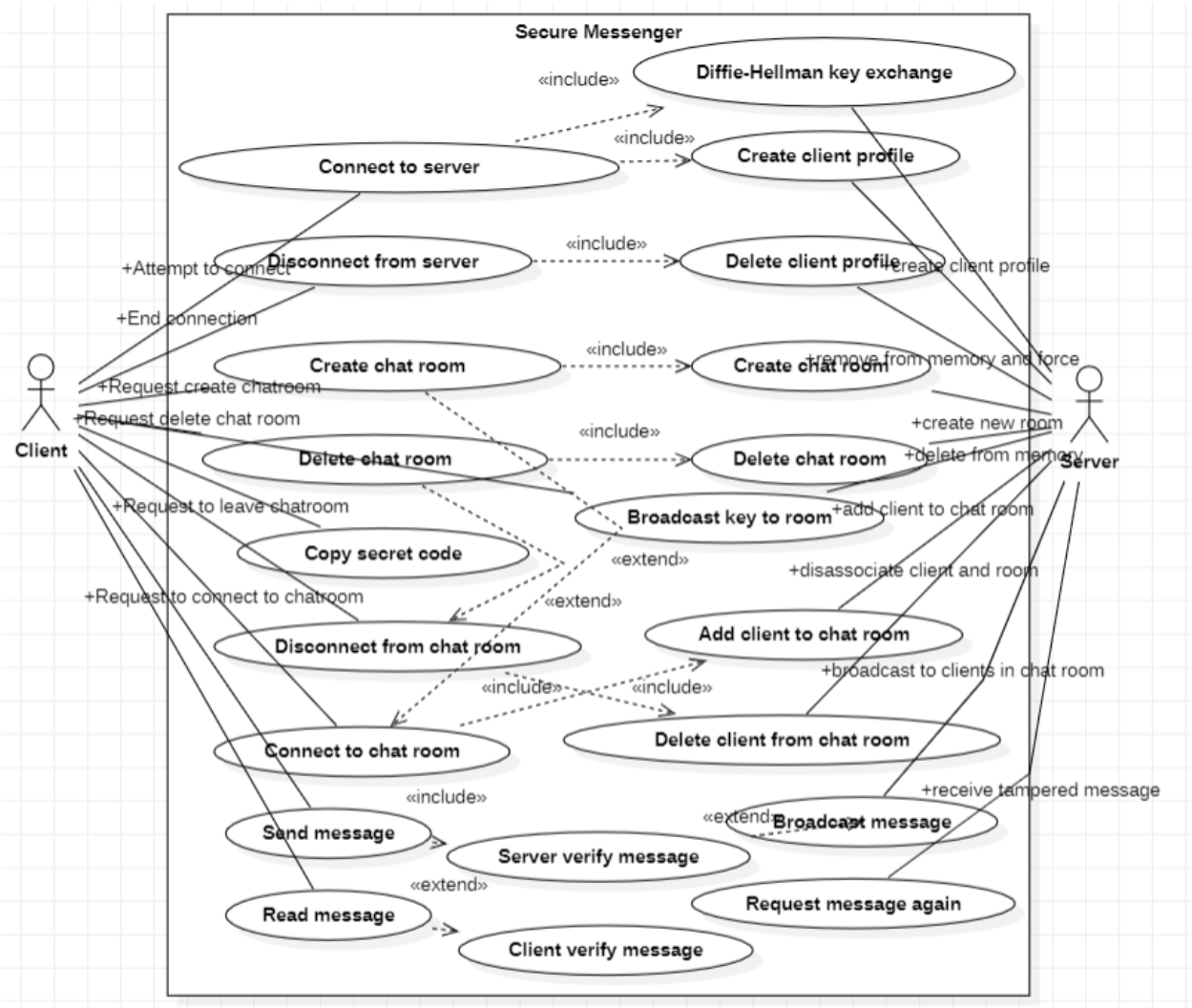
LMMT-SecureDove

CPTS 428 Fall 2023
Professor Haipeng Cai

Logan Kloft
Manjesh Puram
Matthew Gerola
Taylor West

Table of Contents

I. Use Case Diagram



II. Use Case Tables

Use Case Name	**Send message**
Actors	User
Preconditions	● Connected to the server ● Joined an existing room ● Gave public key to server ● Another user is in the room
Goal	Send a confidential message to another user
Scenario	1. The user wants to send a message to someone in the room 2. The user types their message in the message box 3. The user presses send, which encrypts the message 4. The encrypted message gets sent to the room
Exceptions	● No one is in the room to receive the message

Use Case Name	**Read message**
Actors	User
Preconditions	● Connected to the server ● Joined a room ● Gave public key to server ● Another user sent a message
Goal	Read a confidential message sent from another user
Scenario	1. The user is in a conversation with another user in the room 2. The other user sends a message 3. The user reads the encrypted message sent from the other user 4. The message gets decrypted to a viewable form
Exceptions	● Message gets lost in transit ● Connection gets disrupted ● No one is in the room to send a message

Use Case Name	**Connect to server**
Actors	User
Preconditions	● Working internet connection ● Public/Private key generated

Goal	Connect to the main server to get in a room to communicate with others and exchange keys with other users.
Scenario	1. The user wants to connect to communicate with another user 2. The user opens the application and presses connect
Exceptions	● The server is down ● No internet

Use Case Name	**Disconnect from server**
Actors	User
Preconditions	● Was connected to the server prior
Goal	Leave the server and break the link.
Scenario	1. The user is done messaging 2. The user clicks disconnect from server 3. The user leaves the server
Exceptions	● Application crashes

Use Case Name	**Connect to chat room**
Actors	User
Preconditions	● Gave public key to server ● Connected to the server ● Chat room is up
Goal	Connect to communicate with another user in a secure environment
Scenario	1. The user is connected to the server 2. The user clicks connect to chat room 3. The user inputs the code 4. The user is connected
Exceptions	● The room doesn’t exist ● Connection interruption

Use Case Name	**Disconnect from chat room**
Actors	User
Preconditions	● Connected to a room prior
Goal	Leave a chat room to possibly join another room

Scenario	1. The user is done talking with the people in the room 2. The user clicks leave room 3. The user now has left the room and can join other rooms
Exceptions	● The server crashes

Use Case Name	**Copy secret code**
Actors	Client (Potentially server if checks if room still exists)
Preconditions	● Client connected to server ● Client inside of a chat room
Goal	Invite other clients to a chat room using a code. Provide a copy function to make sharing the code both easier and faster.
Scenario	1. Client wants to invite other clients 2. Client navigates to desired chat room 3. Client clicks copy button by room code 4. Client shares code simply by pasting it and sending it over a different platform
Exceptions	● Chat room does not exist

Use Case Name	**Create client profile**
Actors	Server, Client
Preconditions	
Goal	Allow the server to track clients while they are connected to facilitate messaging capabilities
Scenario	1. Client requests a connection to the server 2. Server accepts connection 3. Server creates a client instance 4. Server responds to client with success
Exceptions	● Not enough memory ● Client already exists

Use Case Name	**Delete client profile**
Actors	Server, Client

Preconditions	
Goal	After a client disconnects, or upon request by client, remove that client's profile from memory.
Scenario	1. A client disconnects from the server
Exceptions	● Client does not exist. Ignore silently

Use Case Name	**Create chat room**
Actors	Server, Client
Preconditions	
Goal	Allow a server to create an instance of a chat room that starts with one client and can expand to include more.
Scenario	1. A client requests the creation of a new chat room 2. The server creates a chat room 3. The server adds the client profile to the chat room 4. The server lets the client know that it has created a chat room
Exceptions	● No memory available ● Client does not exist

Use Case Name	**Delete chat room**
Actors	Server, Client
Preconditions	
Goal	When a chat room is not in use anymore the chat room should be deleted to save resources and protect integrity of any information whether it is considered useful or not
Scenario	1. All clients leave the chat room
Exceptions	● Chat room does not exist ● Client still in chat room

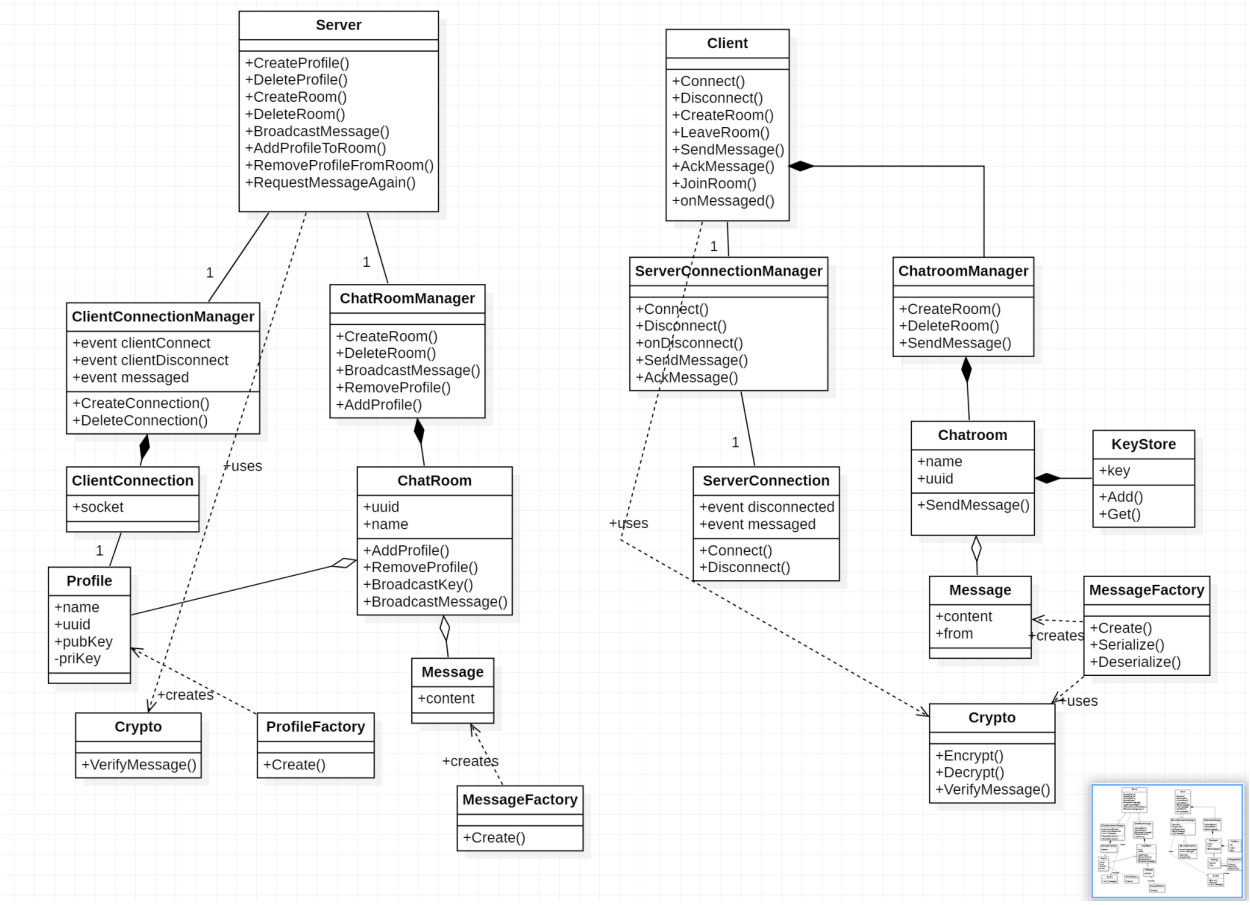
Use Case Name	**Broadcast key to room**
Actors	Client, Server

Preconditions	
Goal	The server can broadcast the key a client is using for encryption to all parties in a chat room that way everyone can read the message from that client.
Scenario	1. Client connects to chat room 2. Server broadcasts client public key to all members in the chat room
Exceptions	

Use Case Name	**Add client to chat room**
Actors	Server, Client
Preconditions	
Goal	Allow clients to join chat rooms so they may chat with other clients.
Scenario	1. Client sends chat room code to server 2. Server adds client profile to chat room 3. Server broadcasts new client public key
Exceptions	

Use Case Name	**Broadcast message**
Actors	Server, Client
Preconditions	
Goal	A client should be able to send messages to everyone. This is already covered. However, on the server side, the server can broadcast a message to everyone in the chat room.
Scenario	1. Client sends message 2. Server receives message 3. Server checks message integrity 4. Server broadcasts message to respective chat room
Exceptions	● Chat room does not exist ● Message integrity bad

III. Class Diagrams



IV. Quality Plan

A. Security Goals

Goal: At every step of message transferral, the message is encrypted. (Confidentiality)
Strategy: Clients encrypt and decrypt messages.

Goal: The message is undecipherable while encrypted. (Confidentiality)
Strategy: Use cryptographically secure encryption techniques

Goal: Prevent a client from joining a chat room they don't belong in. (Confidentiality)
Strategy: Make every chatroom require a long code to join.

Goal: Prevent an attacker from mimicking somebody else (CIA)
Strategy: Make profiles unique. Associate a profile with a socket, as opposed to an ip for example.

B. Security Metrics

Metric: Server downtime (Availability)
Measurement: Have the server log every 5 minutes. Then in the case of the server going down, one can look at the last log to calculate how much time the server was down.

Metric: Message success rate (Availability)
Measurement: When a client sends a message to the server, or the server sends a message to a client, the initiator will increment a counter for messages sent. And then also increment a counter for ACKs if the receiver sends back an acknowledgement message. The message success rate is the quotient between messages sent and ACKs received. This can be logged on a configurable interval.

Metric: Encrypted Message
Measurement: True if intercepted message is encrypted, false otherwise.