CMP2204 – Term Project Report

Peer-to-Peer Secure Chat Application

1. Development Environment

This project was developed on Windows 11 operating systems using Visual Studio Code and terminal environments. The Python version used throughout the development was 3.10+. The required library for DES encryption, pyDes, was installed via pip.

2. Challenges and Solutions

Several challenges were encountered during the development and testing phases of the project:

- **Local IP Visibility Issues:** While the system worked flawlessly on one group member's device (Gokhan), the same code failed to detect local ports or establish proper IP-based peer discovery on another member's PC. This was likely caused by Windows Firewall or system-level socket blocking. This was resolved after allowing Python through the firewall and testing with different routers.
- **Unlogged Incoming Messages:** Initially, the system correctly logged outgoing messages to `history.txt` but failed to log received messages into `chatlog.txt`. This was later fixed by adding correct message handling and write operations in `chat_responder.py`.
- **Router Broadcasting Issues:** The system initially failed on a specific modem/router setup. After switching to a different router provided by another team member, broadcasting and peer discovery worked as expected.
- **Cross-Team Compatibility (Encryption):** Initially, secure communication failed between different teams. Once `pyDes` and `base64` were integrated correctly into both teams' implementations, secure messaging between independent systems was achieved.

3. Team Members and Contributions

Gokhan Yavuz – Network Communication and Peer Discovery Lead

- Developed and tested `service_announcer.py` and `peer_discovery.py`
- Implemented UDP broadcast and handled `users.json`
- Conducted Wireshark analysis of the discovery phase
- Wrote technical reasoning for the first two issues in the troubleshooting table

Raouf Alipour – Security and Encryption Lead

- Coded Diffie-Hellman key exchange, DES encryption, and base64 encoding in `chat_initiator.py` and `chat_responder.py`
- Tested secure messaging flow
- Collected and interpreted secure messaging screenshots for Wireshark Analysis
- Wrote and reviewed the "Security Mechanisms" section of the README

Ahmet Erbey – UI, Documentation, and Testing Lead

- Designed the CLI interface menus in `chat_initiator.py`
- Maintained `history.txt`, `chatlog.txt`, and validated user scenarios
- Created and structured the complete `README.md`
- Wrote command summary and scenario documentation
- Organized and labeled final test and demo screenshots

Note: All team members contributed across all components. While roles were assigned for coordination, each member actively participated in the learning, coding, testing, and debugging processes throughout the project.