# Secure Client-Server Chat Project Report

## 1. Project Overview

This project implements a secure client-server chat system with certificate-based authentication, encrypted communication, replay/tamper protection, and evidence collection for InfoSec assignment submission.

## 2. Environment Setup

- **OS:** Linux
- **Python:** 3.x (venv used)
- **Dependencies:** `cryptography`, `pymysql`, `python-dotenv`
- **Database:** MySQL
- **Version Control:** Git/GitHub

## 3. Git & Repository Initialization

- Initialized local git repository
- Added `.gitignore` for Python, venv, and evidence files
- Pushed code to remote GitHub repository

## 4. Certificate Generation & Validation

- Generated CA, server, and client certificates using OpenSSL
- Implemented certificate exchange and validation in server/client
- Tested with valid and invalid (self-signed) certificates

## 5. MySQL Setup & Schema

- Created MySQL user and database
- Imported schema from `mysql/schema.sql`:

  ```
  CREATE TABLE users (
    id INT PRIMARY KEY AUTO_INCREMENT,
    email VARCHAR(255) UNIQUE,
    username VARCHAR(255) UNIQUE,
    salt BINARY(16),
    pwd_hash BINARY(32)
  );
  ```

- Verified user registration and login

## 6. Server & Client Implementation

- `src/server.py` and `src/client.py`: Main logic for registration, login, and encrypted control channel

- `src/server_chat.py` and `src/client_chat.py`: Chat logic with transcript and receipt
- Used AES for encryption, DH for key exchange, RSA for signatures

## 7. Security Features

- **Certificate validation:** Only CA-signed certs accepted
- **Encrypted channel:** AES session key via DH
- **Replay/tamper protection:** Monotonic sequence numbers, signature verification
- **Transcript/receipt:** Server generates signed receipt with transcript hash

## 8. Testing & Evidence Collection

- **Replay/tamper tests:** Used `tools/replay_send.py` and `tools/tamper_send.py` with crafted packets
- **Transcript/receipt verification:** Used `tools/verify_receipt.py` to validate server receipt
- **Bad/expired cert test:** Replaced client cert with self-signed, server rejected connection
- **Evidence:**
  - Saved tcpdump PCAP (`evidence_chat.pcap`)
  - Collected MySQL user data
  - Saved server/client outputs and screenshots

## 9. Final Verification & Submission

- All code and evidence committed and pushed to GitHub
- Verified all assignment requirements are met

## 10. File Structure

securechat-skeleton-main/ |– app/ | |– client.py | |– server.py | |– common/ | |– crypto/ | |– storage/ |– certs/ |– mysql/ |– scripts/ |– tests/ |– tools/ |– transcripts/ |– .gitignore |– requirements.txt |– README.md

## 11. References

- Python cryptography docs
- MySQL documentation
- Assignment instructions

---

**Prepared by:** FazalZamanKhan **Date:** November 14, 2025