Peer Review Report — Group 97

Reviewer: Mahrin Alam Mahia

Student_ID: a1957342

Overview

The review examined Group 97's C++20 implementation of the SOCP v1.3 secure chat protocol.

Testing was performed on Linux (WSL 2) using CMake 3.20 and OpenSSL 3.0+, with local compilation and execution of both server and client binaries.

Functionality was verified across one-server and multi-server scenarios following the provided README instructions.

Source-level inspection was conducted on main.cpp, server.cpp, client.cpp, crypto.cpp, database.cpp, websocket.cpp, and utils.cpp to confirm security correctness and protocol compliance.

Automated analysis was supported by clang-tidy, cppcheck, and OpenSSL API inspection for potential cryptographic misuse.

Findings

Intentional backdoors

Vulnerability	Description	Impact
Introducer Trust Bypass	bootstrap.json entries are accepted without signature validation; the server implicitly trusts remote introducers.	Allows malicious introducer to inject false routing information.
Replay / Deduplication Gap	SOCP v1.3 requires message ID tracking, but server.cpp does not persist a seen-ID cache across sessions.	Replayed messages are re-accepted after restart (DoS potential).
Missing Signature Check on Bootstrap Join	During server-to-server handshake, key exchange is accepted without verifying the introducer's RSA signature.	Can enable spoofed server identities.

File Transfer Timing Window	Files are hashed after write completion in client.cpp; no per-chunk verification.	Short window where partial corruption goes undetected.

Minor Weakness

- Thread Detachment in websocket.cpp: Detached threads lack join synchronization; could leave dangling connections on shutdown.
- Lack of Input Sanitization: No explicit JSON schema validation in client.cpp; malformed payloads may throw runtime exceptions.
- Error Logging: std::cerr is used without timestamp or severity levels, making debug auditing harder.
- No TLS Transport: All encryption is application-level; WebSocket traffic is unencrypted on port 8080.

Recommendations

- Validate bootstrap introducers with digital signatures or trusted CA list; Prevents spoofed server joins.
- Persist message ID cache across sessions to stop replays; Maintains protocol integrity.
- Add chunk-level hash verification during file transfer; Ensures end-to-end integrity.
- Integrate optional TLS (wss://) layer for transport security; Protects metadata from MITM exposure.
- Introduce structured logging with timestamps and severity; Improves debugging and incident response.