Peer Review Report — Group 101

Reviewer: Mahrin Alam Mahia

Student_ID: a1957342

Overview

Group 101's chat application is a well-structured implementation of the SOCP v1.3 protocol, employing RSA-OAEP for encryption and RSA-PSS for message integrity.
The system correctly supports direct and broadcast communication and includes asynchronous networking with websockets.

The accompanying PoCs successfully demonstrate known security flaws intentionally left in the code for assessment purposes.
Overall, the system operates reliably under normal single-server conditions; however, during a multi-server test, a runtime disconnection (ConnectionClosedOK: 1000 OK) was encountered, which represents a robustness issue rather than a backdoor.
This condition arises when a client attempts to send after a clean connection close, indicating a missing lifecycle or reconnection check.

1. Findings

1.1 Intentional backdoors

Vulnerability	Description	Impact
Weak RSA Key Acceptance (poc_weak_rsa.py, crypto.py)	Accepts 1024-bit keys without enforcement of minimum size.	Allows feasible key factorization and message forgery.
Presence / Routing Poisoning (poc_presence_poison.py, server.py)	Server trusts user advertisements without signature validation.	Enables session impersonation and message misdelivery.
Replay Attack Gap (poc_replay.py, tables.py)	No replay-ID cache or timestamp validation.	Permits duplicate packet replay and confusion of state.
File Integrity Exposure (poc_file_no_check.py, client.py / server.py)	Chunks written before final SHA-256 verification.	Transient window where tampered data exists on disk.

1.2 Operational Weakness

- Broad Exception Catching: except Exception: used extensively in client.py, db.py, and tables.py, suppressing important errors.

- Global Variables in Client State: Shared globals (user_name, ws_connection, privkey) reduce thread safety and clarity.
- Encoding Omissions: File operations lack encoding="utf-8", creating platform dependency risks.
- Runtime Connection Error: ConnectionClosedOK (1000) when sending after server shutdown — a stability bug, not a security flaw.

2. Static Analysis Report

Pylint (project files)

Overall score: 7.94 / 10. Numerous missing-docstring and too-many-branches/statements indicators in client handlers. Repetitive broad-exception-caught and open without encoding. Minor import-order and naming style issues in tables.py and db.py.

```
socp\tables.py:27:4: W0602: Using global for 'seen_ids' but no assignment is done (global-variable-
not-assigned)
socp\tables.py:61:15: W0718: Catching too general exception Exception (broad-exception-caught)
socp\tables.py:79:11: W0718: Catching too general exception Exception (broad-exception-caught)

-----------------------------------
Your code has been rated at 7.94/10
```

Code is functional and readable; most warnings are stylistic or refactor suggestions rather than security risks.

Bandit (project files)

When .venv included

```
70              print("[setup] Generating 1024-bit test key (weak)...")
71              priv = rsa.generate_private_key(public_exponent=65537, key_size=1024)
72              pub = priv.public_key()

--------------------------------------------------

Code scanned:
        Total lines of code: 711807
        Total lines skipped (#nosec): 9

Run metrics:
        Total issues (by severity):
                Undefined: 0
                Low: 3288
                Medium: 219
                High: 23
        Total issues (by confidence):
                Undefined: 0
                Low: 28
                Medium: 53
                High: 3449
```

When .venv excluded

Result: Production code paths contain no medium or high severity issues; Bandit assessment = Low risk. Excluding the venv/site-packages yields an accurate project-only view.

```
   Location: .\poc\poc_weak_rsa.py:71:15
70              print("[setup] Generating 1024-bit test key (weak)...")
71              priv = rsa.generate_private_key(public_exponent=65537, key_size=1024)
72              pub = priv.public_key()

--------------------------------------------------

Code scanned:
        Total lines of code: 2595
        Total lines skipped (#nosec): 0

Run metrics:
        Total issues (by severity):
                Undefined: 0
                Low: 51
                Medium: 1
                High: 0
        Total issues (by confidence):
                Undefined: 0
                Low: 0
                Medium: 1
                High: 51
```

Recommendations

- Enforce RSA key length ≥ 4096 bits in crypto.py; Eliminates intentional weak-key backdoor.
- Verify signatures on USER_ADVERTISE and routing events; Stops presence/routing poisoning.
- Introduce replay-ID cache and timestamp filtering; Mitigates duplicate message injection.
- Validate file chunks pre-write or use temp storage pending hash check; Ensures transfer integrity.
- Narrow exception types and add error logging; Improves observability and security monitoring.
- Guard client send operations after socket close or auto-reconnect; Fixes ConnectionClosedOK robustness issue.