Reviewer: Samin Yeasar Seaum
Student ID: a1976022
Course: Secure Programming (SOCP Phase 3 — Testing and Peer Review)

Peer Review Report — Group 43 ('Vulnerable')

1. Executive Summary

This peer review evaluates Group 43's implementation of the Secure Online Chat Protocol (SOCP). The review focuses on code maintainability, security practices, and architectural design while acknowledging two intentionally included educational backdoors. The feedback below is constructive and aimed at helping the group improve their secure programming techniques.

2. Backdoors

Two intentional vulnerabilities were found. They are documented here:

Backdoor 1 – Unsigned USER_ADVERTISE Messages: The implementation applies presence advertisements to server state without first verifying the sender's transport signature. This demonstrates the importance of ordering verification before state mutation in gossip-based systems.

Backdoor 2 – Weak Public Key Acceptance: User public keys are accepted without enforcing RSA-4096 modulus length. This intentionally highlights the need for strict key-strength validation during registration and storage.

3. Code Quality and Maintainability

The project is modular and readable, with clear separation between client, server, introducer, and cryptographic utilities. However, a few handlers are lengthy and could be split into smaller functions to improve testability and clarity.

4. Security Review (excluding intentional backdoors)

Aside from the documented educational weaknesses, the codebase shows generally sound security practices but would benefit from the following improvements:

- Replace broad try/except/pass blocks with explicit exception handling and logging.
- Default the introducer bind to 127.0.0.1 rather than 0.0.0.0, unless external exposure is required and controlled.
- Ensure logging does not include sensitive material (private keys or raw plaintext payloads).
- Add explicit key-format and strength checks in the normal registration flow.

5. Compliance with SOCP v1.3

Below is a mapping of the project to key SOCP v1.3 requirements and recommended minimal actions where compliance is incomplete.

• RSA Key Policy (RSA-4096 required): Partial — RSA primitives used, but registration lacks explicit modulus enforcement. Action: enforce key_size >= 4096 on key import and reject otherwise.

• Signature Enforcement on Server Frames: Partial — handlers exist, but USER_ADVERTISE updates state before verification in the backdoor variant. Action: verify 'sig' before any state change; add unit tests for unsigned rejects.

• Bootstrap & Introducer Pinning: Mostly followed — introducer flow implemented. Action: ensure pinned pubkeys are used to verify SERVER_WELCOME and validate assigned_id.

• Loop/Replay Suppression: Unclear — explicit seen-IDs cache not obvious. Action: add seen_ids LRU/TTL cache to avoid duplicate forwarding.

• Mandatory Client Commands and Heartbeats: Mostly present. Action: add end-to-end tests for /list, /tell, /all, /file and heartbeat timeout behavior.

6. Testing Methodology

Testing was performed using Pylint (score: 7.75/10), Bandit scans, manual code inspection, and dynamic observation in an isolated Docker sandbox. All tests were non-destructive and focused on verifying state transitions, logging, and handler behavior.

7. Recommendations (actionable & supportive)

1. Enforce strict key validation on import and reject keys below 4096 bits.
2. Ensure signature verification precedes any state mutation in all server handlers.
3. Replace silent exception handlers with explicit logging and fail-closed behavior.
4. Introducer binding should default to localhost; require explicit flag/documentation for external exposure.
5. Refactor complex handlers into smaller units and add docstrings and type hints.
6. Integrate Pylint and Bandit into CI and add unit/integration tests for the core flows listed in the compliance section.

8. Strengths

- Clear modular architecture separating responsibilities.
- Appropriate use of RSA primitives and signing where applied.
- Backdoors intentionally included for learning and are documented.
- Readable code style and sensible helper utilities for crypto.

9. Areas for Improvement

- Improve exception handling to enforce fail-closed semantics.
- Add modulus checks and duplicate-registration checks in registration flows.
- Reduce cyclomatic complexity in message handlers.
- Improve inline documentation and comments for security-critical logic.

Appendix — Tool Evidence Summary

Pylint Summary:

- Score: 7.75/10
- Issues: Missing docstrings, complex handler functions.

Bandit Summary:

- B104: Hardcoded bind to all interfaces (introducer.py).
- B110: Broad try/except/pass (server.py).

Manual Review:

- Two educational backdoors verified and documented.
- No additional unintentional vulnerabilities identified in the static scan and manual review.

Conclusion

Group 43's submission is a strong implementation of SOCP and meets many of the project's learning goals. Implementing the recommended minimal changes — key validation, signature ordering, explicit error handling, and simple deduplication — will align the project more closely with SOCP v1.3 and increase robustness for real-world scenarios.