Secure LAN Communication System Project Presentation

Good morning distinguished judges and professors,

Today I will present our team's secure local area network communication system. This system implements DES encryption algorithm and consists of client/server modules supporting group chat, private messaging, and broadcast communication modes.

For connection establishment, clients create TCP connections through socket API. The server adopts multi-threaded architecture where each new connection spawns an independent processing thread. Implemented using Winsock library on Windows, the server listens on port 5000 by default, with clients connecting via 127.0.0.1 for local testing.

Our communication protocol follows "TYPE|TARGET|CONTENT" format with six command types including login, group creation, and message broadcasting. Before encryption, we apply PKCS#7 padding for 8-byte alignment, use CBC mode for enhanced security, and implement HMAC-SHA1 for integrity verification.

The DES encryption module features complete implementation: 64-bit plaintext undergoes initial permutation before being split into 32-bit halves. Sixteen Feistel network rounds process the data followed by final permutation to generate ciphertext. Key generation involves 56-bit permutation and 16 circular shifts to produce 48-bit subkeys. Each encryption round expands 32-bit data to 48-bit, XORs with subkey, then compresses through S-box substitution.

Notably, the Feistel network processes half-data per round using XOR operations for reversible encryption. This symmetric structure enables decryption by reversing subkey order. Our CBC implementation uses initialization vectors to ensure different ciphertexts for identical plaintexts.

For multi-device support, the server maintains group-member mappings using hash tables, with each client connection handled by dedicated threads. Client state machines track joined/owned groups for permission management. When sending group messages, the server iterates through member lists for message routing.

Testing procedures include: compiling/running the server, connecting multiple client instances, and verifying functions through group creation/joining/messaging commands. Encryption tests use standard vectors to validate DES implementation, including ciphertext comparison with known keys.

Live demonstration shows encrypted network packets becoming completely unreadable while decryption accurately restores original messages. The system supports 20 concurrent stable connections with <100ms end-to-end latency. Stress testing confirms single-server capacity for 200 concurrent connections, suitable for small-medium office environments.

Future plans include implementing RSA key exchange for hybrid encryption and optimizing thread pool management for high-concurrency scenarios. This concludes our project presentation. Thank you for your attention and we welcome any questions.