1 Project Documentation

1.1 Assumptions

The team assumes that the communicating parties have already exchanged their public keys securely through an out-of-band mechanism. This implies that before initiating any communication using the chat program, users must have a trusted method of sharing their public keys with each other to avoid the complexities and potential vulnerabilities associated with a public key infrastructure (PKI). The system also assumes that each party's private keys and any derived session keys are securely stored and managed, preventing unauthorized access or leakage.

Furthermore, it is assumed that the adversary has significant capabilities, including intercepting, modifying, or replaying messages exchanged between the communicating parties (Man-in-the-Middle attacks). The adversary has access to the network and can perform active attacks, such as injecting messages or impersonating a participant. However, it is also assumed that the adversary cannot break the cryptographic primitives (AES-256, HMAC-SHA-256, Diffie-Hellman key exchange) within a feasible time frame using current computational resources. The security of the random number generator used by the cryptographic library (OpenSSL) is also assumed to be strong and unpredictable.

1.2 Claims

Integrity: Each message includes a Message Authentication Code (MAC) generated using HMAC-SHA-256. This ensures that any modification to the message during transit will be detected by the recipient. The integrity of the message is verified before decryption, preventing attacks that modify the ciphertext to induce predictable changes in the plaintext.

Confidentiality: In the code, all messages exchanged between the communicating parties are encrypted using AES-256 in CBC mode. This ensures that an adversary who intercepts the messages cannot read their contents. Session keys derived from the Diffie-Hellman key exchange are used for encryption. These keys provide perfect forward secrecy, meaning that if a session key is compromised, it does not affect the security of past sessions.

Mutual Authentication: Both communicating parties authenticate each other using their respective public keys. This prevents an adversary from impersonating one of the parties. The Diffie-Hellman key exchange ensures that both parties agree on a shared secret, which is used to derive session keys for encryption and MAC generation. This mutual agreement ensures that both parties are legitimate and in possession of their private keys.

In the case of malicious communicating party: As mentioned above, in the case of a malicious communication party performing Man-in-the-Middle attacks, they can decrypt, read, and modify messages. Hence, it would be good to integrate certificate verification to ensure the authenticity of public keys exchanged.

2 Possible Attack

One potential attack on the system is a Man-in-the-Middle (MitM) attack. In this scenario, if an attacker can intercept Alice's (Client A's) ephemeral key and instead send Bob (Client B) their own key, the attacker can effectively control the communication. This allows the attacker to decrypt messages from Alice, modify them if desired, and then encrypt and send the modified messages to Bob using the attacker's own ephemeral key. Bob, believing he is communicating securely with Alice, is actually exchanging messages with the attacker.

Perfect forward secrecy (PFS) aims to protect past communication sessions from being decrypted even if the current session keys are compromised. This is achieved by generating new keys for each session. However, PFS alone cannot protect against MitM attacks if the initial key exchange is compromised. In the case of an MitM attack, the attacker can establish separate keys with both Alice and Bob, decrypting and re-encrypting messages in real-time without either party realizing the breach.

To mitigate this risk, the Triple Diffie-Hellman (3DH) key exchange protocol can be employed. The 3DH protocol involves three separate Diffie-Hellman exchanges to provide stronger security properties. Specifically:

- Each party generates a long-term key pair and an ephemeral key pair for each session.
- The parties exchange their public keys, both long-term and ephemeral.
- Each party computes three shared secrets using their private keys and the received public keys:
  - One shared secret using their long-term private key and the other party's ephemeral public key.
  - Another shared secret using their ephemeral private key and the other party's long-term public key.
  - A third shared secret using their ephemeral private key and the other party's ephemeral public key.
- These shared secrets are then combined (e.g., using a hash function) to derive the final session key.

By combining these multiple key exchanges, 3DH ensures that an attacker would need to compromise multiple keys simultaneously to succeed in a MitM attack. The mutual authentication step, where both parties authenticate each other using their long-term public keys, further mitigates the risk by ensuring that the parties are indeed communicating with the intended counterpart and not an attacker.

Thus, the 3DH protocol not only provides perfect forward secrecy but also significantly enhances the security against MitM attacks, ensuring a more robust and secure communication channel.