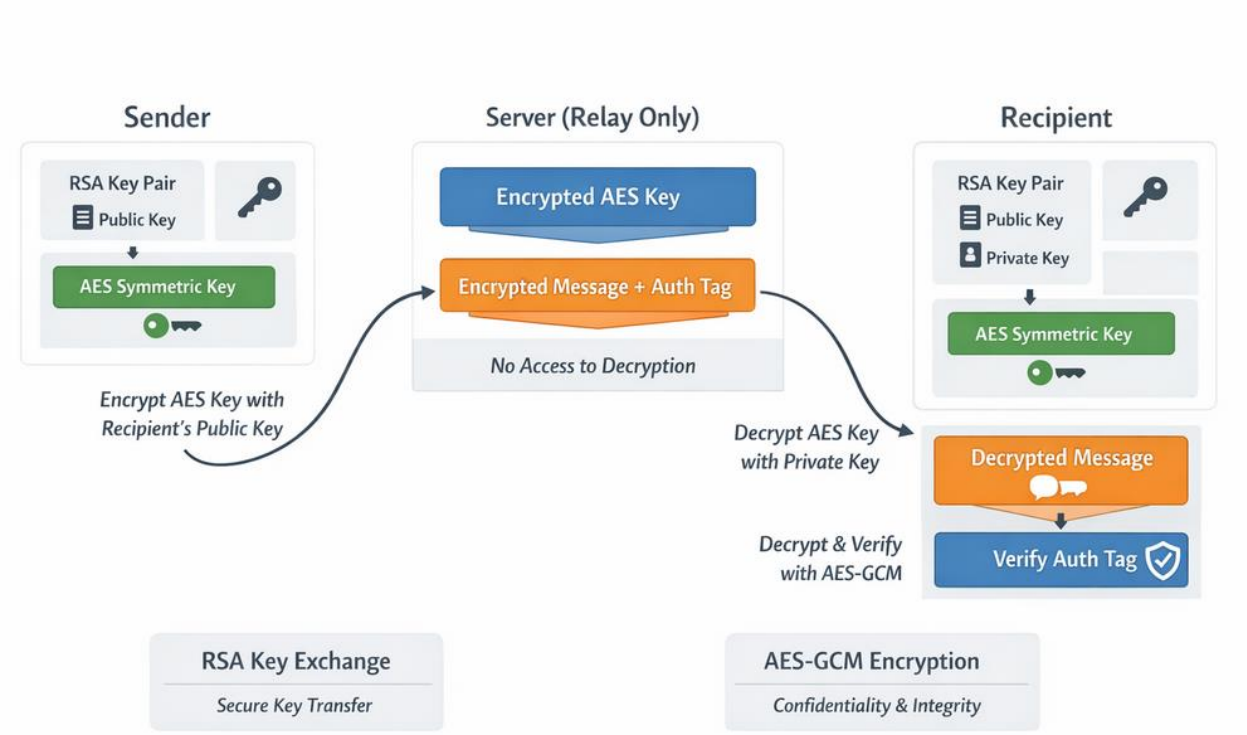
Security Model for the Application



Key Generation and Exchange

- Each client generates:
  - A public/private key pair (RSA) for asymmetric encryption
  - A shared symmetric key (AES) for message encryption

- The symmetric key is encrypted with the recipient's public RSA key and sent via the server.

- The server only relays encrypted keys and messages it cannot decrypt any content.

Message Encryption

- Messages are encrypted using (AES-GCM).
    - AES provides confidentiality.
    - GCM mode provides both encryption and integrity.

- Workflow for sending a message:
    - Sender uses AES-GCM with the shared symmetric key to encrypt the message.
    - AES-GCM generates an authentication tag to ensure the message hasn't been tampered with.
    - The encrypted message + tag is sent to the server, which relays it to the recipient.
    - Recipient decrypts the message using the shared symmetric key and verifies the authentication tag.

Server Role

- Relays encrypted messages and key exchanges.
- Does not have access to plaintext messages or symmetric keys.

Security Benefits

- End2End Encryption: Only the intended recipient can decrypt the message.
- Integrity and Authenticity: AES-GCM ensures messages cannot be tampered with undetectably.
- Performance: AES-GCM is efficient for encrypting message payloads compared to asymmetric encryption for every message.

Trade-offs and Weaknesses

- Key Management: Loss of private keys means permanent loss of messages.
- Server Trust: Server is trusted for key delivery; if compromised, attackers could perform key substitution attacks unless keys are verified.
- Group Chats: Requires securely distributing AES keys to all group members.

Possible Improvements

- Use Forward Secrecy with ephemeral symmetric keys to prevent past message compromise if keys are leaked.
- Implement digital signatures for stronger authenticity verification.
- Use client-side key verification to prevent man-in-the-middle attacks during key exchange.
