University of Bahrain

Department of Computer Science

ITCS411 – Cryptography and Computer Security

2023/2024 First Semester

Secure Chat Application

Done by:

Maha Mohammed	202002565
Musherah Moqbel	202002276
Noor Jaafar	202007922
Maram Hussain	202007755
Farha Alsadah	202003353
Maryam Emadudin	202002021

Secure Chat Applications

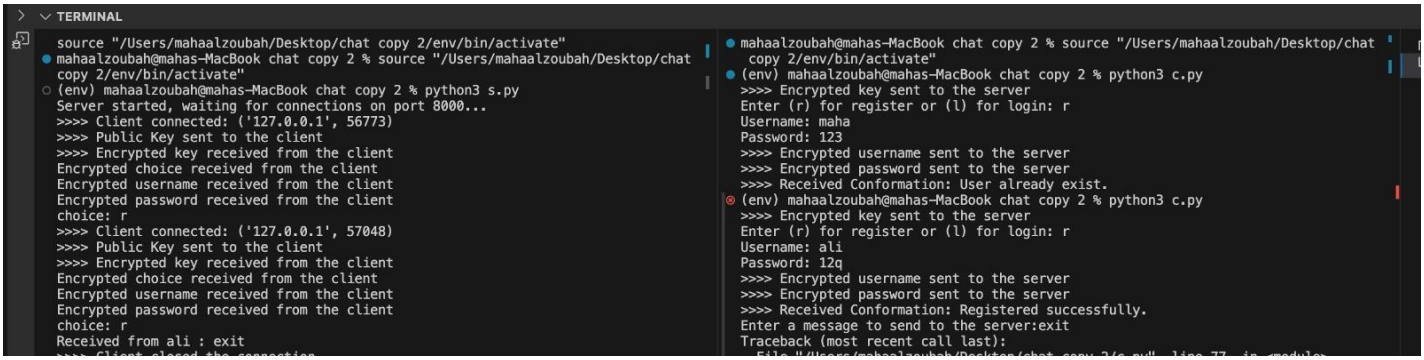
The Secure Chat Application serves as an innovative solution for facilitating secure communication over a network, emphasizing the importance of confidentiality and integrity in message exchange. This report delves into the application's design, implementation, and the security algorithms employed to ensure robust protection against potential threats.

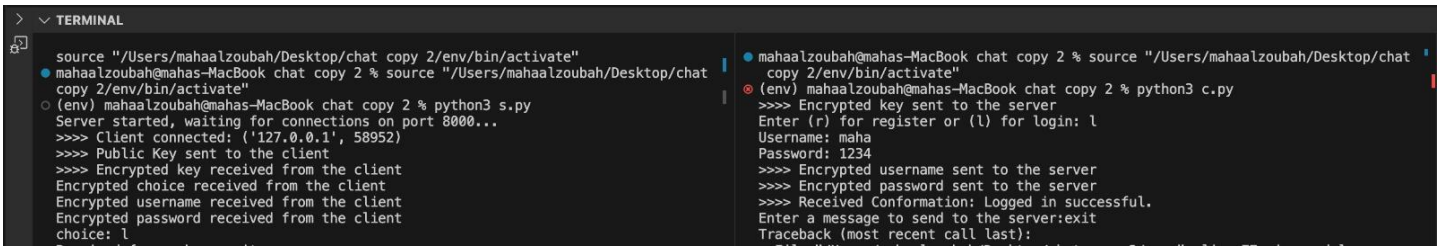
The primary objective of the Secure Chat Application is to provide users with a platform for confidential and secure communication. Leveraging Python's socket programming and network concepts, the application establishes a client-server architecture to facilitate communication channels. Key functionalities include a secure user registration process, end-to-end message encryption and link scanning for shared URLs.

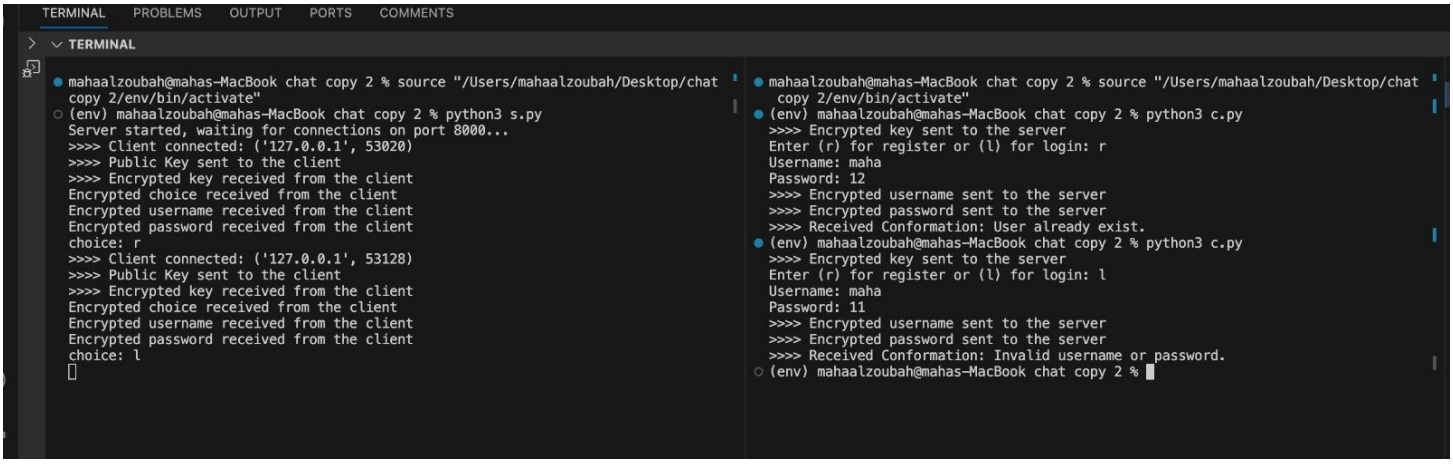
The RSA algorithm, named after its architects Rivest, Shamir, and Adleman, is an essential component of secure communication. RSA is an asymmetric encryption method that employs two keys: public and private. The public key is freely distributed, allowing anybody to encrypt messages, whereas the private key is kept secret and only needed for decoding. RSA is frequently used in secure chat applications for the initial key exchange, ensuring a secure and private mechanism for users to transfer secret keys and establishing a secured channel for ongoing conversation.

The Advanced Encryption Standard (AES) is a symmetric encryption method designed to secure actual communication once the first key exchange is complete. In contrast to RSA, AES employs a single secret key for both encryption and decryption. AES is well-suited for real-time communication in secure chat apps due to its efficiency and speed. Its strength comes from its capacity to process fixed-size data blocks, which adds a strong layer of confidentiality and integrity to the continual transmission of information.

In today's linked web ecosystem, link scanning is a critical feature that improves chat applications' security posture. Recognizing the pervasive threat posed by rogue URLs and phishing efforts, link scanning software automatically scans shared links for potential threats. The application can proactively alert users or restrict access to connections identified as potential threats to security by scanning the destination for recognized viruses, phishing sites, or other dangerous information. This safeguard ensures that users can interact with shared material without being exposed to malicious activity.


TERMINAL
source "/Users/mahaalzoubah/Desktop/chat copy 2/env/bin/activate"
mahaalzoubah@mahas-MacBook chat copy 2 % source "/Users/mahaalzoubah/Desktop/chat copy 2/env/bin/activate"
(env) mahaalzoubah@mahas-MacBook chat copy 2 % python3 s.py
Server started, waiting for connections on port 8000...
>>>> Client connected: ('127.0.0.1', 56773)
>>>> Public Key sent to the client
>>>> Encrypted key received from the client
Encrypted choice received from the client
Encrypted username received from the client
Encrypted password received from the client
choice: r
>>>> Client connected: ('127.0.0.1', 57048)
>>>> Public Key sent to the client
>>>> Encrypted key received from the client
Encrypted choice received from the client
Encrypted username received from the client
Encrypted password received from the client
choice: r
Received from ali : exit
mahaalzoubah@mahas-MacBook chat copy 2 % source "/Users/mahaalzoubah/Desktop/chat copy 2/env/bin/activate"
(env) mahaalzoubah@mahas-MacBook chat copy 2 % python3 c.py
>>>> Encrypted key sent to the server
Enter (r) for register or (l) for login: r
Username: maha
Password: 123
>>>> Encrypted username sent to the server
>>>> Encrypted password sent to the server
>>>> Received Conformation: User already exist.
(env) mahaalzoubah@mahas-MacBook chat copy 2 % python3 c.py
>>>> Encrypted key sent to the server
Enter (r) for register or (l) for login: r
Username: ali
Password: 12q
>>>> Encrypted username sent to the server
>>>> Encrypted password sent to the server
>>>> Received Conformation: Registered successfully.
Enter a message to send to the server:exit
Traceback (most recent call last):


TERMINAL
source "/Users/mahaalzoubah/Desktop/chat copy 2/env/bin/activate"
mahaalzoubah@mahas-MacBook chat copy 2 % source "/Users/mahaalzoubah/Desktop/chat copy 2/env/bin/activate"
(env) mahaalzoubah@mahas-MacBook chat copy 2 % python3 s.py
Server started, waiting for connections on port 8000...
>>>> Client connected: ('127.0.0.1', 58952)
>>>> Public Key sent to the client
>>>> Encrypted key received from the client
Encrypted choice received from the client
Encrypted username received from the client
Encrypted password received from the client
choice: l
mahaalzoubah@mahas-MacBook chat copy 2 % source "/Users/mahaalzoubah/Desktop/chat copy 2/env/bin/activate"
(env) mahaalzoubah@mahas-MacBook chat copy 2 % python3 c.py
>>>> Encrypted key sent to the server
Enter (r) for register or (l) for login: l
Username: maha
Password: 1234
>>>> Encrypted username sent to the server
>>>> Encrypted password sent to the server
>>>> Received Conformation: Logged in successful.
Enter a message to send to the server:exit
Traceback (most recent call last):


TERMINAL
PROBLEMS
OUTPUT
PORTS
COMMENTS
TERMINAL
mahaalzoubah@mahas-MacBook chat copy 2 % source "/Users/mahaalzoubah/Desktop/chat copy 2/env/bin/activate"
(env) mahaalzoubah@mahas-MacBook chat copy 2 % python3 s.py
Server started, waiting for connections on port 8000...
>>>> Client connected: ('127.0.0.1', 53020)
>>>> Public Key sent to the client
>>>> Encrypted key received from the client
Encrypted choice received from the client
Encrypted username received from the client
Encrypted password received from the client
choice: r
>>>> Client connected: ('127.0.0.1', 53128)
>>>> Public Key sent to the client
>>>> Encrypted key received from the client
Encrypted choice received from the client
Encrypted username received from the client
Encrypted password received from the client
choice: l
mahaalzoubah@mahas-MacBook chat copy 2 % source "/Users/mahaalzoubah/Desktop/chat copy 2/env/bin/activate"
(env) mahaalzoubah@mahas-MacBook chat copy 2 % python3 c.py
>>>> Encrypted key sent to the server
Enter (r) for register or (l) for login: r
Username: maha
Password: 12
>>>> Encrypted username sent to the server
>>>> Encrypted password sent to the server
>>>> Received Conformation: User already exist.
(env) mahaalzoubah@mahas-MacBook chat copy 2 % python3 c.py
>>>> Encrypted key sent to the server
Enter (r) for register or (l) for login: l
Username: maha
Password: 11
>>>> Encrypted username sent to the server
>>>> Encrypted password sent to the server
>>>> Received Conformation: Invalid username or password.
(env) mahaalzoubah@mahas-MacBook chat copy 2 %