**COMP. 5610 Computer and Network Security Project**

**Encrypted Overlay Network**

**Submitted by:**

**Surendar Mourougan (**[**Surendar\_Mourougan@student.uml.edu**](mailto:Surendar_Mourougan@student.uml.edu)**) – Project Coordinator**

**Geetha Sri Harsha Mattaparthy (geethasaisriharsha\_mattaparthy@student.uml.edu)**

**Bhavagna Lanka (**[**Bhavagna\_Lanka@student.uml.edu**](mailto:Bhavagna_Lanka@student.uml.edu)**)**

**Abstract**

Encrypted overlay networks play a crucial role in modern communication systems, providing a secure and reliable platform for data transmission over potentially insecure networks. This paper presents an overview of encrypted overlay networks, discussing their definition, design principles, implementation considerations, and security features. By encrypting communication channels and implementing authentication mechanisms, encrypted overlay networks offer protection against various threats, ensuring the confidentiality, integrity, and authenticity of transmitted data. Through a combination of encryption protocols, authentication mechanisms, and secure communication channels, encrypted overlay networks enhance the overall security posture of networked environments, making them an indispensable component of modern communication infrastructures.

**Overview of the project**

In our project, we implemented an encrypted overlay network to facilitate secure communication between clients over potentially insecure networks. The main components of our project include a network server and multiple clients, each identified by a unique name.

To ensure the security of communication within the network, we incorporated several network security building blocks, including:

1. **Encryption:** We utilized encryption protocols to secure the communication channels between clients and the network server. This ensured that data transmitted between clients and the server remained confidential and protected from eavesdropping.

2. **Authentication:** We implemented authentication mechanisms to verify the identity of clients when they register with the network server. This prevented unauthorized access to the network and ensured that only legitimate clients could participate in the communication process.

3. **Secure Communication Channels:** We established secure communication channels between clients and the network server using encryption protocols such as SSL/TLS. This protected data transmission from interception and tampering by malicious actors.

4. **Key Management**: We employed key management practices to securely generate, distribute, and manage cryptographic keys used for encryption and decryption. This ensured the confidentiality and integrity of data exchanged within the network.

Overall, by incorporating these network security building blocks into our project, we created a robust and secure communication infrastructure that protected the confidentiality, integrity, and authenticity of data exchanged between clients within the encrypted overlay network.

**My Contribution**

* Designed the architecture of the encrypted overlay network, outlining the components, communication protocols, and security mechanisms.
* Conducted thorough testing of the implemented system to ensure its functionality, security, and scalability.
* Addressed any bugs, vulnerabilities, or performance issues identified during testing.
* Made sure that the network pipe is not broken for incoming client connections.

**Future Implementation**

1. Enhancing Security Measures: We aim to strengthen the security of our network by implementing additional security measures such as intrusion detection systems, firewalls, and access control policies to protect against potential threats and attacks.

2. Optimizing Performance: We plan to optimize the performance of our network by streamlining communication protocols, minimizing latency, and improving scalability to accommodate a larger number of clients and higher data throughput.

3. Integrating Advanced Features: We will explore the integration of advanced features such as end-to-end encryption, multi-factor authentication, and secure messaging protocols to further enhance the security and functionality of our network.

Overall, our goal is to continue building upon our existing encrypted overlay network to create a robust and reliable communication infrastructure that meets the evolving security needs of modern networks.

**System Framework and Programming Platform**

Our system framework is built using Python, a high-level programming language known for its simplicity and readability. Python offers a wide range of libraries and modules that facilitate rapid development and prototyping, making it well-suited for building network applications.

For the network communication aspect, we utilize Python's built-in `socket` library, which provides a low-level interface for network communication over TCP/IP protocols. This allows us to create network servers and clients that can establish connections, send and receive data packets, and manage network communication effectively.

In addition to the `socket` library, we also leverage other Python libraries such as `threading` for concurrent execution of multiple tasks, `argparse` for command-line argument parsing, and `time` for managing time-related operations.

For encryption and authentication, we rely on cryptographic libraries such as `cryptography` in Python, which provides robust implementations of encryption algorithms, key management, and cryptographic protocols. This allows us to incorporate encryption mechanisms such as symmetric encryption (e.g., AES) and asymmetric encryption (e.g., RSA) to secure data transmission and authentication protocols (e.g., RSA signatures) to verify the identity of clients.

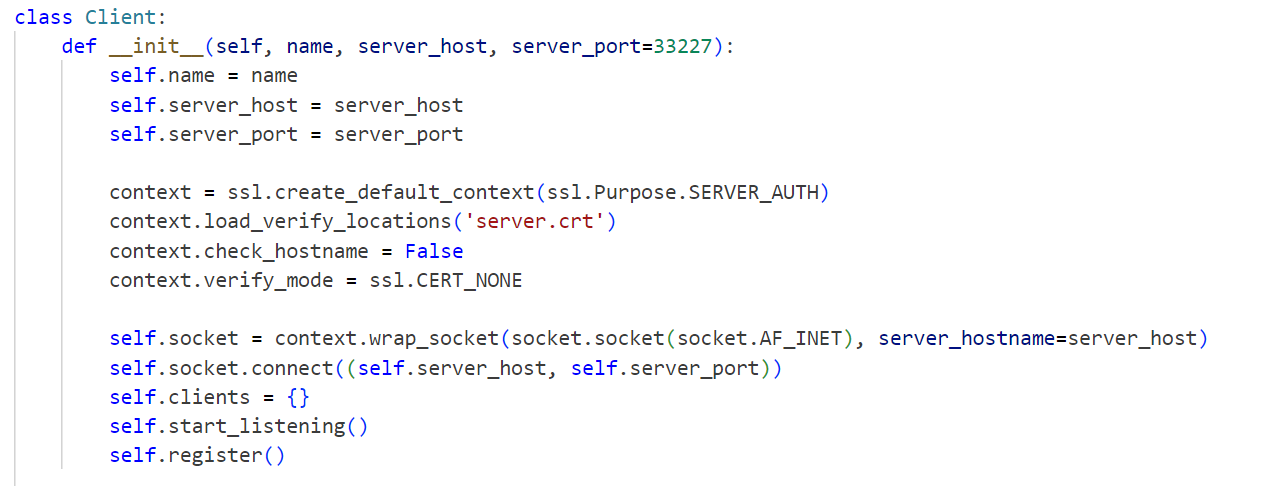
Overall, our system framework is built on Python, leveraging its versatility, simplicity, and extensive libraries to create a secure and efficient network communication infrastructure.

**Sample codes**

**Server.py**

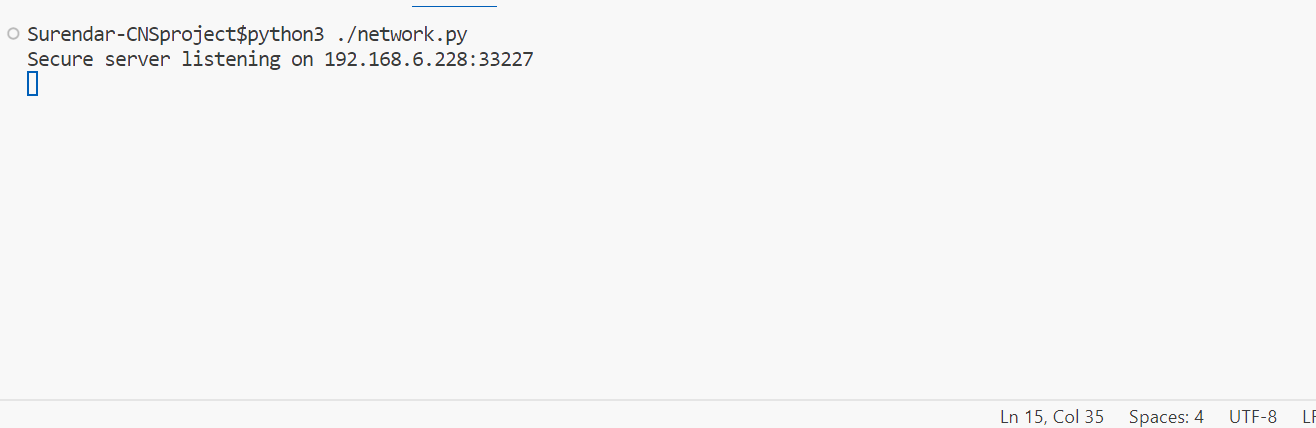
****

**Client.py**

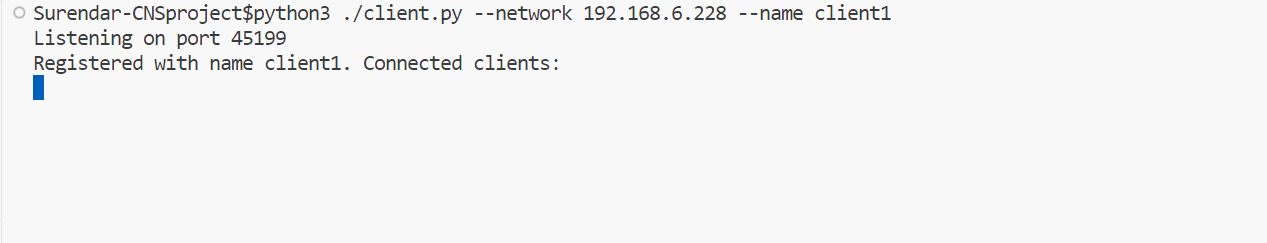
****

**Output samples**

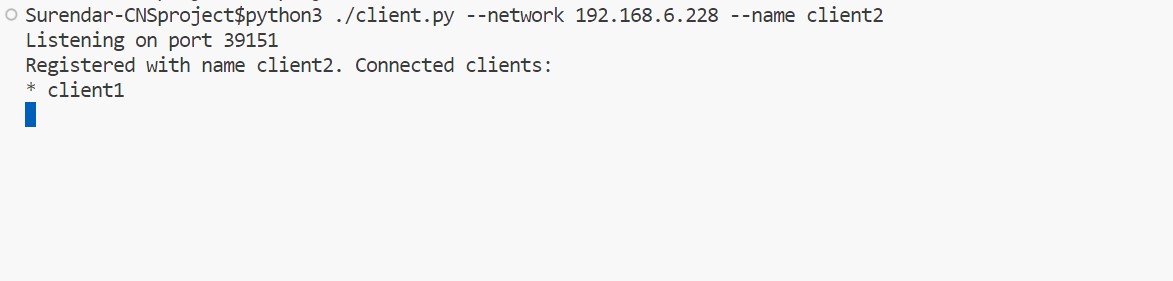
The network is started using this command and the server is listening on 192.168.6.228

****

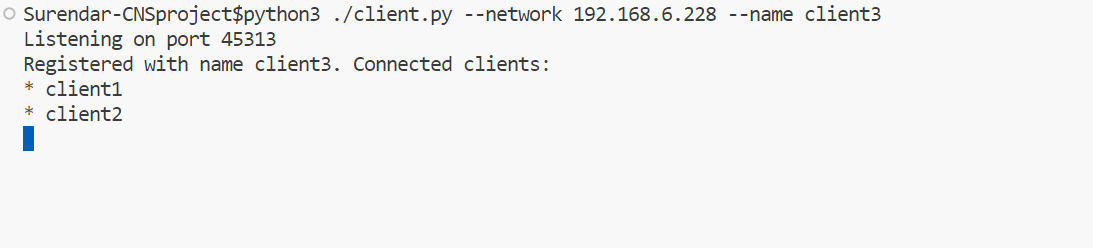
Client1 is initiated and it is joined to the network 192.168.6.228

****

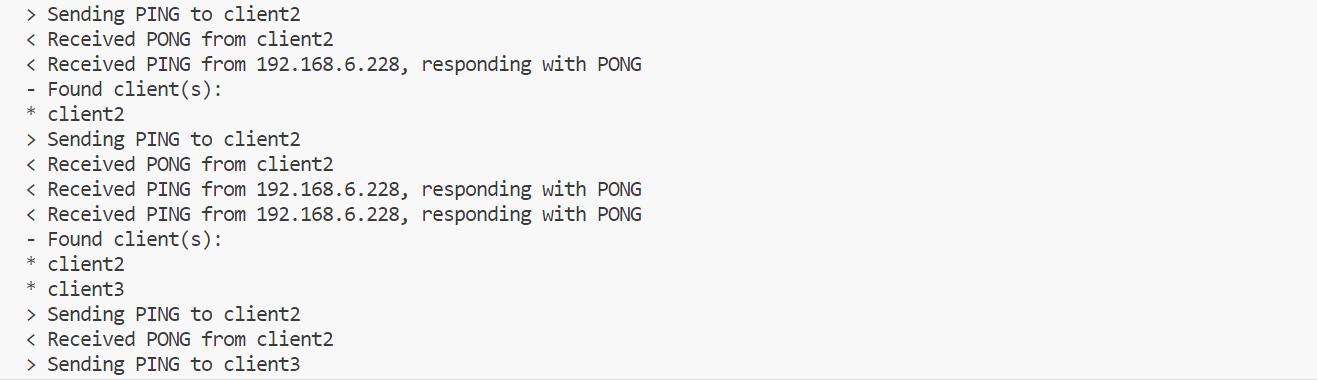
Client 2 is initiated and it is joined to the network 192.168.6.228. We could also see the registered clients in the network.

****

Client 3 is initiated and it is joined to the network 192.168.6.228. We could also see the registered clients in the network.

****

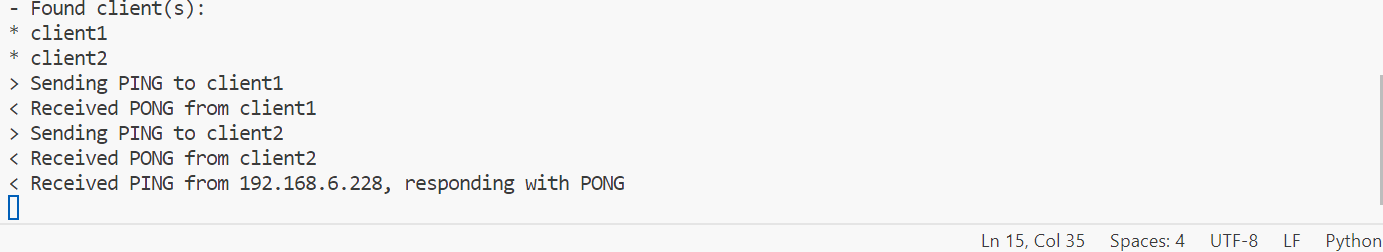
This is the terminal of client 1. We could see the registered clients in the network. Then it sends a PING request to all the clients present in the network and receives a PONG response.



This is the terminal of client 2. We could see the registered clients in the network. Then it sends a PING request to all the clients present in the network and receives a PONG response.



This is the terminal of client 3. We could see the registered clients in the network. Then it sends a PING request to all the clients present in the network and receives a PONG response.



This is the network terminal. Here we can see all the registered clients in the network.

