Secure File Transfer with Encryption

1. Introduction

This project implements a secure file transfer protocol over a TCP/IP network using SSL/TLS encryption. The goal is to demonstrate a solid understanding of network programming, encryption protocols, certificate management, and performance analysis by building a secure and cross-platform client-server file transfer system.

2. System Design

2.1. Components

TLS-Enabled Client (`client.py`)
TLS-Enabled Server (`server.py`)
Plain TCP Client (`default_client.py`) for performance comparison
Plain TCP Server (`default_server.py`) for performance comparison

2.2. Protocol Commands

`SEND filename` — Client sends a file to server

`GET filename` — Client requests a file from server

3. Implementation Details

Programming Language and Version: Python 3

3.1. Libraries Used:

For client and server files

socket

ssl

os

time

For the certificate

ipaddress

datetime

cryptography

3.2. Security Features

TLS encryption via `ssl.SSLContext`

Certificate-based authentication (self-signed X.509)

Input validation to avoid directory traversal attacks

Handling network disconnects and malformed requests

3.3. Error Handling

Catches SSL/TLS errors
Detects and rejects unsafe filenames
Handles EOF detection for file transfer termination
Incomplete transmission or disconnect is logged

3.4. Platform Support

Tested on Windows 10 and Ubuntu Linux

4. User Manual

Installation:

Install Python 3.7+

Place server.crt and server.key in project folder

Run server:

python server.py

Run client:

python client.py

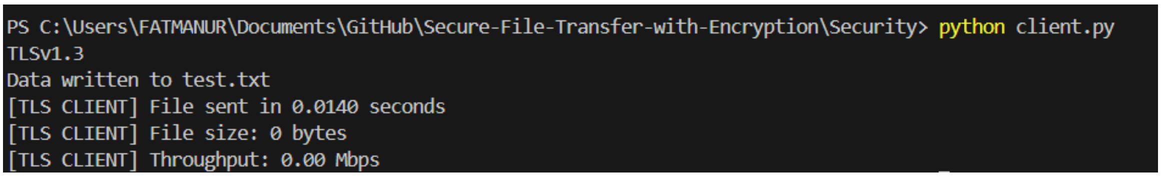
Usage Instructions:

To send a file: set MODE = 'SEND' in client.py

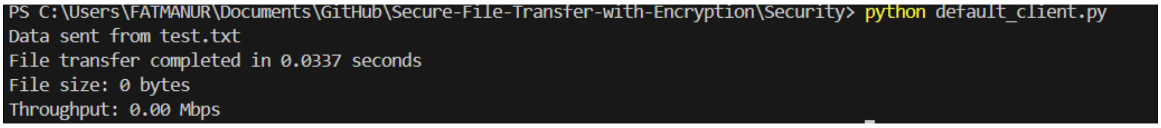
To receive a file: set MODE = 'RECIEVE' in client.py

5. Performance Analysis

Encrypted TCP file is 2.4 times slower than plain TCP file.



```
PS C:\Users\FATMANUR\Documents\GitHub\Secure-File-Transfer-with-Encryption\Security> python client.py
TLSv1.3
Data written to test.txt
[TLS CLIENT] File sent in 0.0140 seconds
[TLS CLIENT] File size: 0 bytes
[TLS CLIENT] Throughput: 0.00 Mbps
```



```
PS C:\Users\FATMANUR\Documents\GitHub\Secure-File-Transfer-with-Encryption\Security> python default_client.py
Data sent from test.txt
File transfer completed in 0.0337 seconds
File size: 0 bytes
Throughput: 0.00 Mbps
```

6. Conclusion

This project demonstrates an end-to-end secure file transfer system using TLS over sockets. It implements encryption, certificate validation, and file operations with input validation and error handling. Performance testing confirms trade-offs between security and speed. Future improvements could include multi-client support, GUI interface, and asymmetric authentication.

7. References

https://docs.python.org/3/library/ssl.html#ssl-sockets
https://cryptography.io/en/latest/fernet/
https://docs.python.org/3/library/os.html#
https://www.youtube.com/watch?v=QhQFEmbRmsY&t=283s