A Project Report on

Encrypted Image Transmission using TCP

21AIE431

Applied Cryptography

Under the guidance of:

Dr Sunil,

Department of CEN,

Amrita School of Computing.

4th October, 2023

Submitted By:

N *Sai Satwik Reddy*	*CB.EN.U4AIE21139*
V Poorna Muni Sasidhar Reddy	*CB.EN.U4AIE21172*
V Venkata Alluri Rohith	*CB.EN.U4AIE21174*
Y Shashank Reddy	*CB.EN.U4AIE21179*

Group - 3

Batch - B

त्यागेनैके अमृतत्वमानशुः

MATA AMRITANANDAMAYI MATH

श्रद्धावान् लभते ज्ञानम्

AMRITA VISHWA VIDYAPEETHAM

ASC, ETTIMADAI, COIMBATORE

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without mentioning about the people whose constant guidance and encouragement made it possible. We take pleasure in presenting before you, our project, which is result of studied blend of both research and knowledge.

We express our earnest gratitude to our internal guide, Dr Sunil, Department of CEN, our project guide, for her constant support, encouragement and guidance. We are grateful for her cooperation and her valuable suggestions.

Finally, we would also like to thank our group mates for the constant encouragement and the team work which was put throughout the project. We could learn about the various concepts in the project with the each other’s help and support.

DECLARATION

We the undersigned solemnly declare that the project report is based on our own work carried out during the course of our study under the supervision of Dr Sunil, Department of CEN. We assert the statements made and conclusions drawn are an outcome of our research work. We further certify that

I. The work contained in the report is original and has been done by us under the general supervision of our supervisor.

II. We have followed the guidelines provided by the university in writing the report.

III. Whenever we have used materials (data, theoretical analysis, and text) from other sources, we have given due credit to them in the text of the report and giving their details in the references.

Nallamilli Sai Satwik Reddy

Vali Poorna Muni Sasidhar Reddy

Venna Venkata Alluri Rohith

Yeturu Shashank Reddy

ABSTRACT

This project presents a secure image transmission system utilizing image cryptography integrated with TCP. Beginning with file preparation, the sender encodes the image into a Base64-encoded string, subsequently encrypting it for enhanced security. Transmission occurs via a TCP connection, followed by decryption and Base64 decoding on the receiver's side. The final step involves reconstructing the original image from the decoded binary data, ensuring a seamless and secure image transfer. Our project has a plethora of real world applications in various domains like remote sensing, secure web browsing etc.

Process:

1. File Preparation:

The sender need to prepare the Image file that needs to be transmitted to binary file.

2. Base64 Encoding:

The binary file is encoded into a Base64-encoded string using the Base64 encoding process.

This creates a text-based representation of the binary file that can be encrypted and transmitted over the TCP connection.

3. Encryption:

Before transmission, the Base64-encoded file is encrypted to enhance security.

We will do the encryption by using the best possible algorithm.

4. TCP Transmission:

The sender establishes a TCP connection with the receiver's system.

Using the established connection, the sender need to transmit the encrypted Base64-encoded file to the receiver.

Receiver's Side :

5. TCP Reception:

When a connection is established, it accepts the incoming data (the encrypted file) from the sender.

6. Decryption:

The received encrypted file is decrypted using the same cryptographic algorithm and the secret decryption key.

Decryption transforms the encrypted text back into its original form, which is the Base64-encoded string.

7. Base64 Decoding:

The decrypted Base64-encoded string is then subjected to Base64 decoding, following the Base64 decoding process described earlier in the report.

This process reconstructs the original binary file from the decoded Base64 string.

8. Image Reconstruction:

Finally, the reconstructed binary data is interpreted as an image file, allowing the receiver to view or utilize the original image.

Encoding and Decoding :

Base64 encoding and decoding are essential techniques in computer science and information technology for converting binary data into a text-based format and vice versa. These techniques are widely used in various applications where data needs to be represented as text for transmission or storage purposes. Unlike encryption, which is designed to protect data confidentiality, Base64 is a reversible transformation that does not involve secret keys or algorithms; instead, its primary purpose is to facilitate data interchange in a readable and standardized manner.

Base64 Encoding Process:

Chunking: Binary data, such as images, files, or binary streams, is divided into fixed-size chunks of 6 bits each. This step is essential because the Base64 encoding scheme relies on 6-bit units.

Character Mapping: Each 6-bit chunk is then mapped to a corresponding character from the Base64 character set. This character set consists of 64 characters, typically including uppercase and lowercase letters (A-Z, a-z), numbers (0-9), and two additional characters often chosen as '+' and '/'. Each character in the set represents a unique 6-bit binary value.

Concatenation: These mapped characters are concatenated together to form a Base64-encoded string. This string can be safely transmitted through channels that may not support binary data transmission, such as email or URLs.

Base64 Decoding Process:

Chunk Separation: To decode a Base64-encoded string, it is first divided into 4-character chunks. These chunks were created during the encoding process.

Reverse Mapping: Each character in the chunk is then mapped back to its original 6-bit binary representation using the inverse mapping of the Base64 character set. This reverses the encoding process and recovers the original binary data.

Reconstruction: The binary representations obtained from the previous step are combined to reconstruct the original binary data, restoring it to its original format.

TCP Transmission:

1. Set up the server:

Before you can send files, you must first configure a server to accept incoming connections.

The server is typically a computer or system that monitors client requests.

2. Initialization of the Server Socket:

The server creates a socket, which is essentially a network communication endpoint.

An IP address and a port number are assigned to the socket. The IP address and port number combination uniquely identifies the server on the network.

3. Listening on a Server Socket:

The server opens the socket and begins listening for client connections.

It goes into a state of inactivity, waiting for connection requests.

4. Client Socket Creation:

A socket is created on the client side.

The IP address and port number of the server to which the client wishes to connect are specified by the client.

5. Making the Connection:

By providing the server's IP address and port number, the client initiates a connection request to the server.

When the server receives the connection request, it accepts it.

Between the client and the server, a network connection is established.

Transferring Files:

After establishing the connection, the server begins reading the file, which must be sent in smaller, more manageable chunks or packets.

These file chunks are sent to the client over the network via the established TCP connection.

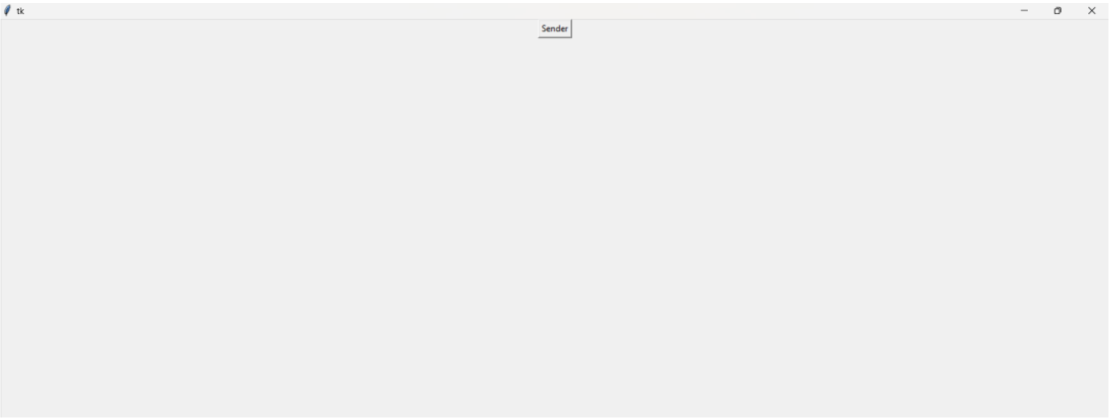
After the entire file has been successfully transferred, both the client and server sockets are closed to release network resources.

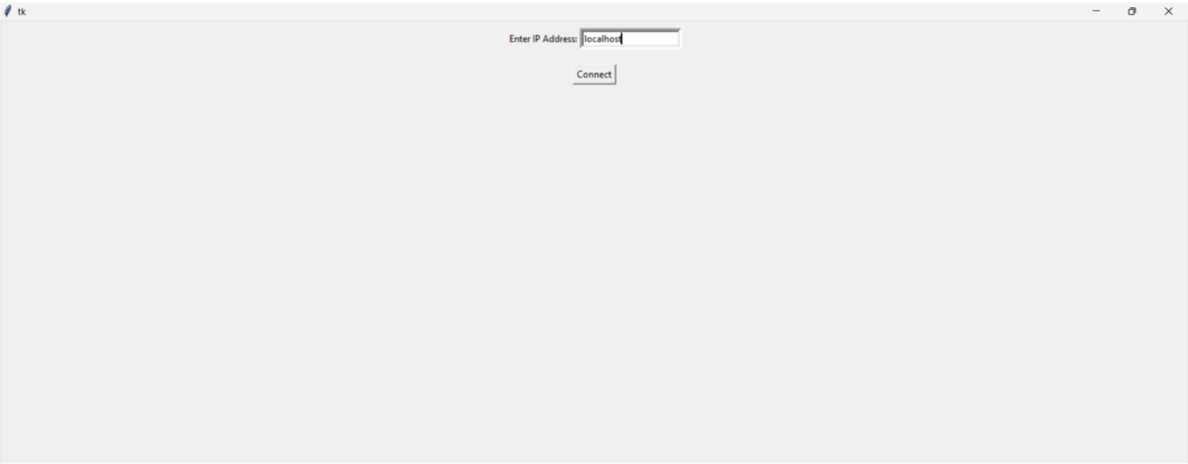
The connection is terminated.

Decryption :

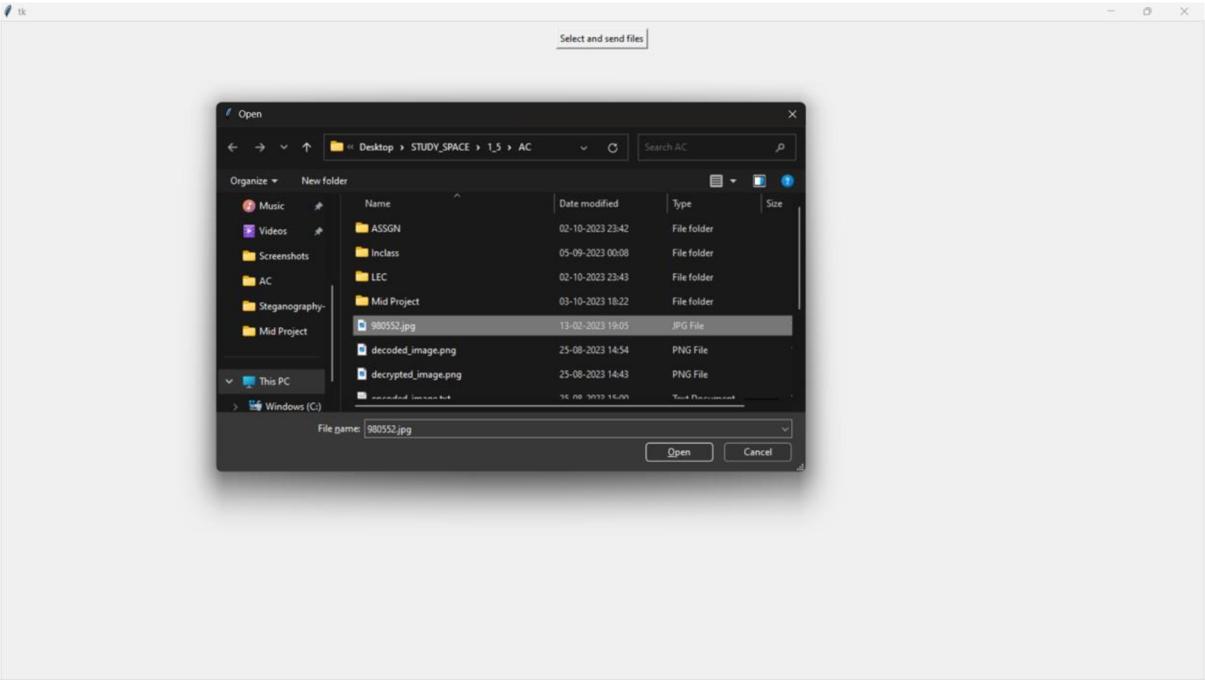
After receiver receives the file it will be decrypted and converted in to original image.

Output:



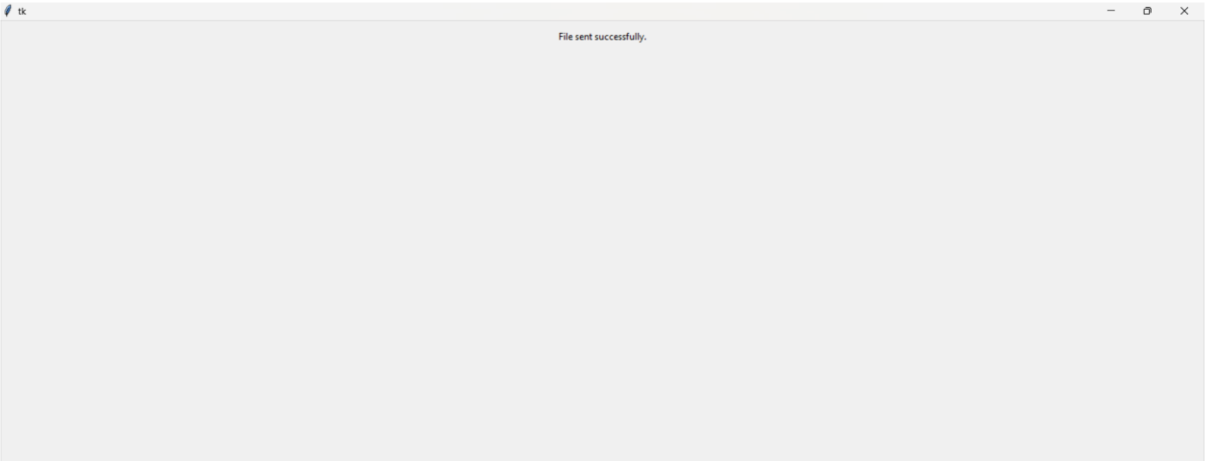


Enter IP Address: localhost

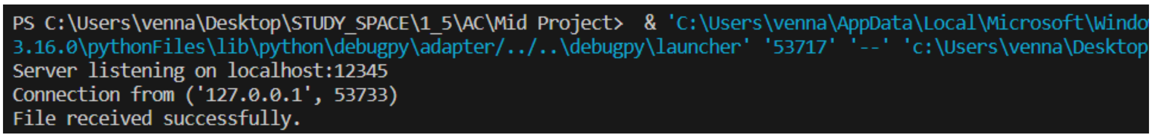


Name	Date modified	Type	Size
ASSGN	02-10-2023 23:42	File folder	
Inclass	05-09-2023 00:08	File folder	
LEC	02-10-2023 23:43	File folder	
Mid Project	03-10-2023 18:22	File folder	
980552.jpg	13-02-2023 19:05	JPG File	
decoded_image.png	25-08-2023 14:54	PNG File	
decrypted_image.png	25-08-2023 14:43	PNG File	
encoded_image.txt	25-08-2023 15:00	Text Document	

File name: 980552.jpg



File sent successfully.



```
PS C:\Users\venna\Desktop\STUDY_SPACE\1_5\AC\Mid Project>  & 'C:\Users\venna\AppData\Local\Microsoft\Windo
3.16.0\pythonFiles\lib\python\debugpy\adapter/../..\debugpy\launcher' '53717' '--' 'c:\Users\venna\Desktop
Server listening on localhost:12345
Connection from ('127.0.0.1', 53733)
File received successfully.
```