

# Computer Networks

Project: Implement a Secure HTTP Server using Steganography

## Group members:-

Asfa Mumtaz (32691)

Ibrahim Usmani (32929)

Syed Wahab Aftab Chishty (32728)

Submitted to: Dr Arslan Ahmad

Submitted on: May 22, 2016

## **Introduction:**

In this project we have designed a secure HTTP server that is able to serve web clients with secure content.

The client sends a filename to the server using the browser,if the file is present server encodes the file into an image using steganography and send the image to the the client.

If the file requested by the client is not present of the server,it generates an error message and sends it to the client.

The client contains a decoder,which decodes the images and stores the text file on the client's computer. The server listen on the port 55555.

HTTP fetches objects from the server and displays it on the client machine.It generates different response and error message in case of presence of absence of objects. The two methods used by http to get requests from the client is the GET and POST method. We have used the same principles to develop our system.

## **Objective:**

The objective of this project is to develop a client server networking model and explore the working of hypertext transfer protocol and the error codes it generates. And to familiarize ourselves with network programming with sockets, client/server systems in python.

Moreover our main target was to develop a secure http server , which we did.

## **Steganography:**

The steganography method used in this project is not the standard LSB steganography but a modified version of it. In standard LSB steganography we edit the last bit of all R,G,B bytes(from pixels), but here we we edited the last 3 bits of R,G,B. In this way we achieved a better storage method that can store more text in smaller images .Each character is encoded in a single pixel using its R,G,B value. The first three bits of the eight character word are stored in the last three bits of the value of R. G stores the next three bits of the character in a similar way. And finally B carries the remain two bits. The method has made no significant/visible changes in our image quality.

## **Overall working:**

- User connects to client from browser.
- User requests file from client
- The clients connects to the server using its ip and port number.
- It sends a request to the server.
- The server fetches this request.
- Server sends an ack if file is available
- After encoding it send the file to the client
- If file is not available it displays 404 error message
- The client decodes the file and stores it on to the machine, as well as sends the image to the browser.

# Problems

Following are the problems encountered during whole of this project:

- Image access and decoding cannot be done in parallel.
- If the text size is really big (i.e. in 100s of MBs) this method will fail and data will be wasted.
- One bit is always wasted in one pixel i.e the last bit of "B" of any Pixel.