**How to execute:**

1. make

2. Server's side: ./pbproxy -k key.txt -l 2222 localhost 22

3. Client's side: ssh -o "ProxyCommand ./pbproxy -k key.txt localhost 2222" localhost

**Sample output:**

**Server:**

aman@Aman-Asus:~/Documents/HW3$ ./pbproxy -k key.txt -l 2222 localhost 22

**Client:**

aman@Aman-Asus:~/Documents/HW3$ ssh -o "ProxyCommand ./pbproxy -k key.txt localhost 2222" localhost

aman@localhost's password:

Welcome to Ubuntu 16.04.3 LTS (GNU/Linux 4.10.0-37-generic x86\_64)

\* Documentation: https://help.ubuntu.com

\* Management: https://landscape.canonical.com

\* Support: https://ubuntu.com/advantage

15 packages can be updated.

14 updates are security updates.

Last login: Wed Nov 8 15:39:56 2017 from 127.0.0.1

aman@Aman-Asus:~$ ls

~ bgoyal.img Downloads output.pcap Public

AlgoHW1.txt Desktop examples.desktop Pictures Templates

amaagarwal.img Documents Music PlayOnLinux's virtual drives Videos

aman@Aman-Asus:~$ exit

logout

Connection to localhost closed.

aman@Aman-Asus:~/Documents/HW3$

**Description:**

The program does the following:

1. Takes the arguments and parses them appropriately.
2. For client side:
3. It creates a socket with socket() call
4. Then it connects the socket with the server’s address with connect() call
5. Then we read data from standard input and write to the socket, and read from socket and write to standard output. (I have used select() to read and write)
6. For server side:
   1. We create a socket with socket() call
   2. Then we bind the socket with bind() on port 2222 (for the above example)
   3. Listen for connections with listen()
   4. Accept incoming connections with accept()
   5. Now we have an active connection and we have the client’s socket fd.
   6. We will now create a new thread and open a connection to port 22
   7. And then we read the data from client’s socket fd and write to sshd socket fd, and read from sshd socket fd and write to client’s socket fd.
7. This way we have a tunnel via 2222, which acts as a reverse proxy and the client only talks to him and it forwards the data to the appropriate socket.