“CRYPTO-SECURED CHATBOX”

A

Project Report

submitted in partial fulfillment of the

requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING

by

Kartik Tyagi	Abhishek Joshi	Sahil Sangwan
500067312	500067827	500068972

Under the guidance of

MR. SANDEEP KUMAR CHAURASIYA
DEPARTMENT OF CYBERNETICS

Department of Cybernetics,
School of Computer Science

CANDIDATE’S DECLARATION

I/We hereby certify that the project work entitled **“ CRYPTO-SECURED CHATBOX "** in partial fulfilment of the requirements for the award of the Degree of BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE AND ENGINEERING with specialization in Open Source and Open Standards and submitted to the Department of Computer Science & Engineering at Center for Information Technology, University of Petroleum & Energy Studies, Dehradun, is an authentic record of my/ our work carried out during a period from **August 2020** to **December 2020** under the supervision of Mr. Sandeep Kumar Chaurasiya, Department Of Cybernatics.

The matter presented in this project has not been submitted by me/ us for the award of any other degree of this or any other University.

Kartik Tyagi	**Abhishek Joshi**	**Sahil Sangwan**
500067312	**500067827**	**500068972**

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Mr. Sandeep Kumar Chaurasiya
Project Guide

ACKNOWLEDGEMENT

We wish to express our deep gratitude to our guide Mr. Sandeep Kumar Chaurasiya, for all advice, encouragement and constant support he has given us throughout our project work. This work would not have been possible without his support and valuable suggestions.

We would like to thank all our friends for their help and constructive criticism during our project work. Finally we have no words to express our sincere gratitude to our parents who have shown us this world and for every support they have given us.

Kartik Tyagi	Abhishek Joshi	Sahil Sangwan
500067312	500067827	500068972

ABSTRACT

Due to increased government surveillance as well as data breaches, end to end encryption has recently received an increasing attention as a way to protect against such threats. The project aims to remove these vulnerabilities and threats by creating and developing a mechanism called secured chat box that is capable of encrypting & decrypting user's text and images and hence protecting user's privacy. The input takes details from the user for his/her general text including the images which in turn will be encrypted using our modified algorithm. The encrypted text and image will be shown as an output which is completely secured and thus protected by various cryptanalyst.

KEYWORDS: Encryption, Decryption, RSA, Socket Programming, AES, DES, Client Server Model

TABLE OF CONTENTS

INTRODUCTION

With the rapid development of mobile phones, laptops, pc’s etc. These devices have become one of the integral part of daily activities. In recent years, chat applications have evolved and made a major change in social media because of their distinctive features that attract audiences. It provides real-time messaging and offers different services including, exchange text messages, images, files and etc. Moreover, it supports cross platforms. There are currently hundred millions of users using chat applications. There are two types of architecture in those applications, client-server and peer-to-peer networks. In a peer-to-peer network, there is no central server and each user has his/her own data storage. On the contrary, there are dedicated servers and clients in a client-server network and the data is stored on a central server. Security and privacy in chat applications have a paramount importance but few people take it seriously. In a test done by the Electronic Frontier Foundation, most of the popular messaging applications failed to meet most security standards. These applications might be using the conversations as an information for certain purposes. Moreover, reading the private conversations is certainly unacceptable in terms of privacy. Most applications only used Transport Layer Security (TLS) for securing channel, the service provider has full access to every message exchanged through their infrastructure. Therefore, these messages can be accessed by attackers. Therefore to maintain protection and privacy, messages should be encrypted from sender to receiver and no one can read messages even the service provider, in addition to protecting the local storage of the device. In this paper, we focus on security, privacy and speed by proposing end-to-end security which ensures only sender and receiver can read messages without a third party. As well as storage protection and fast transfer of messages between the parties.

The main contributions of this project are the following:

- ✓ Propose client-server chatroom.
- ✓ Secure key exchange, then calculate the session key.
- ✓ Secure exchange of end-to-end messages.

LITERATURE REVIEW

Title	Link	Author	Remarks
Cryptography	https://www.pdfdrive.com/serious-cryptography-a-practical-introduction-to-modern-encryption-e183556059.html	Jean-Philippe Aumasson	This practical guide to modern encryption breaks down the fundamental mathematical concepts at the heart of cryptography without shying away from meaty discussions of how they work.We learn about authenticated encryption, secure randomness, hash functions, block ciphers, and public-key techniques such as RSA and elliptic curve cryptography.
RSA	https://www.geeksforgeeks.org/rsa-algorithm-cryptography/?ref=rp	Mohit Gupta	This post defines the coding of RSA and implementation.
Algorithms used in now-a-days	https://www.geeksforgeeks.org/encryption-its-algorithms-and-its-future/	Jash Kothari	It explains about RSA, AES and DES algorithms.
RSA (cryptosystem)	https://en.wikipedia.org/wiki/RSA_(cryptosystem)	Wikipedia	It is one of the first public key cryptosystems and is widely used for secure data transmission. A user of RSA creates and then publishes a public key based on two large prime numbers, along with an auxiliary value. The prime numbers must be kept secret. Anyone can use the public key to encrypt a message, but only someone with knowledge of the prime numbers can decode the message. Breaking RSA encryption is known as the RSA problem. Whether it is as difficult as the factoring problem is an open question. There are no published methods to defeat the system if a large enough key is used.

PROBLEM STATEMENT

To build a secured chat box in order to protect user's data in form of text by using algorithm for encryption-decryption.

OBJECTIVES

- To create a chatbox with secured communication having end to end encryption.
- To enable/ensure confidentiality and security for user data.
- To implement data integrity so as to prevent the alteration by any means.

  - ✓ **ACHIEVED**
    - Creation of chatroom
    - Multiple client handling(Basic)
    - Encryption, Decryption of messages.
    - Optimization of code

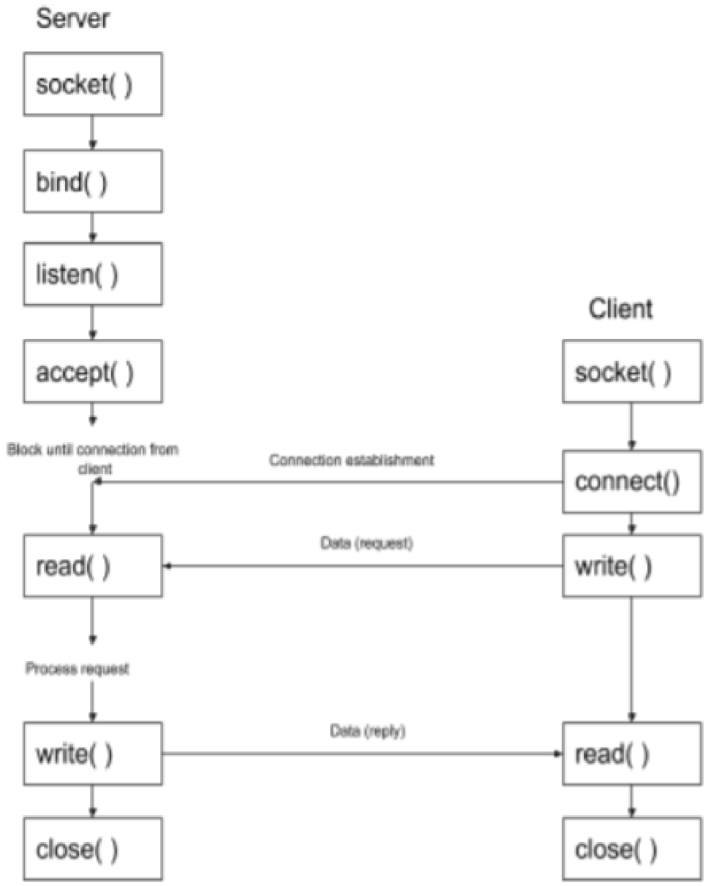
METHODOLOGY

To satisfy above objectives we will follow agile methodology throughout this project. This method is both iterative and incremental that is good for a library as it will keep the library up-to-date.

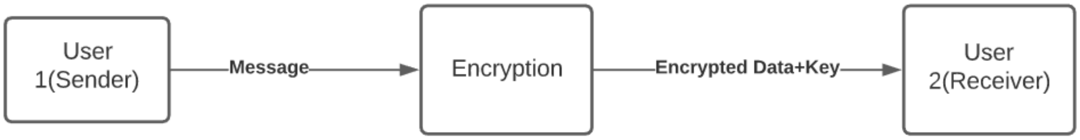
During the development of the library, It is divided based on modules making sure they are loosely coupled with each other (making it flexible for future updates).These modules are coded parallel to each other with a checkpoint of integration so that if we require to add some new feature we can directly work on that particular module to fulfill future requirements.

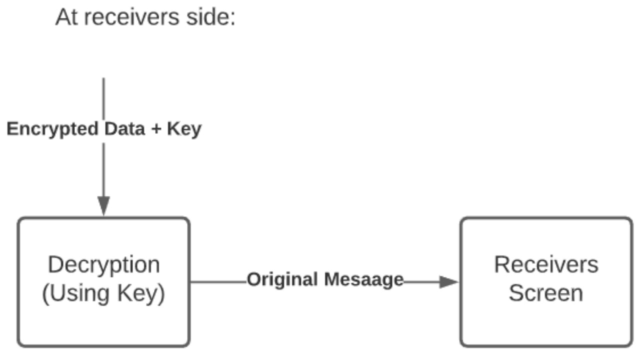
FLOW CHART

Phase 1: Developing chatbox using socket programming, and c.



Phase 2: Implementing encryption and decryption.





ALGORITHM’s USED

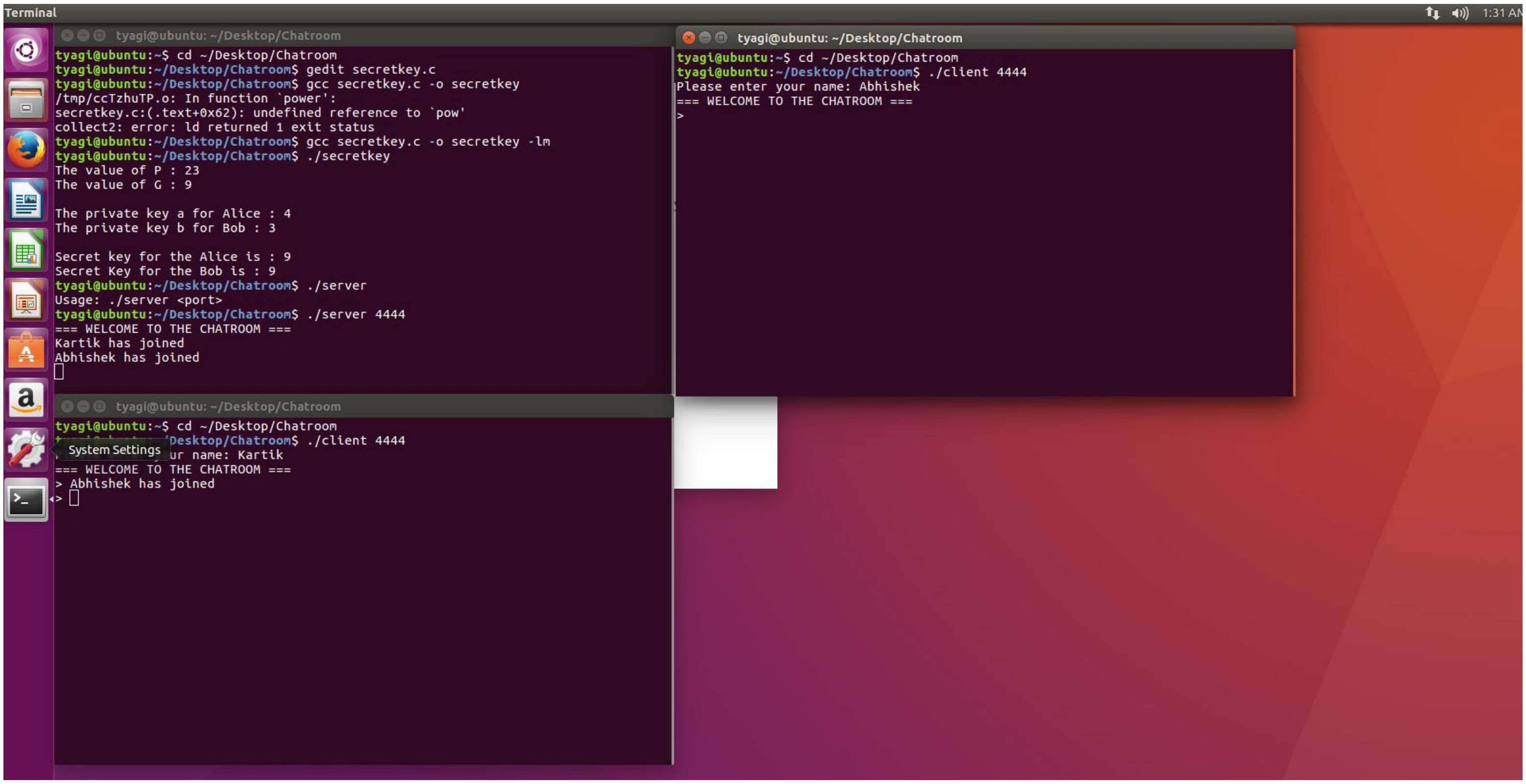
Chatroom : is created using socket programming. Socket programming is a way of connecting two nodes on a network to communicate with each other. One socket(node) listens on a particular port at an IP, while other socket reaches out to the other to form a connection. Server forms the listener socket while client reaches out to the server.

Algorithm for TCP client

- Create socket,connectSocket
- Do an active connect specifying the IP address and port number of server
- Read and Write Data Into
  connectSocket to Communicate with server
- Close connectSocket

Algorithm for TCP server

```
• Create socket (serverSocket)
• Bind socket to a specific port where clients can contact you
• Register with the kernel your willingness to listen that on socket for client to contact you
• Loop
  Accept new connection (connectSocket)
  Read and Write Data Into connectSocket to
      Communicate with client
  Close connectSocket
  End Loop
• Close serverSocket
```



```
tyagi@ubuntu: ~/Desktop/Chatroom
tyagi@ubuntu:~$ cd ~/Desktop/Chatroom
tyagi@ubuntu:~/Desktop/Chatroom$ gedit secretkey.c
tyagi@ubuntu:~/Desktop/Chatroom$ gcc secretkey.c -o secretkey
/tmp/ccTzhuTP.o: In function `power':
secretkey.c:(.text+0x62): undefined reference to `pow'
collect2: error: ld returned 1 exit status
tyagi@ubuntu:~/Desktop/Chatroom$ gcc secretkey.c -o secretkey -lm
tyagi@ubuntu:~/Desktop/Chatroom$ ./secretkey
The value of P : 23
The value of G : 9

The private key a for Alice : 4
The private key b for Bob : 3

Secret key for the Alice is : 9
Secret Key for the Bob is : 9
tyagi@ubuntu:~/Desktop/Chatroom$ ./server
Usage: ./server <port>
tyagi@ubuntu:~/Desktop/Chatroom$ ./server 4444
=== WELCOME TO THE CHATROOM ===
Kartik has joined
Abhishek has joined
```

```
tyagi@ubuntu: ~/Desktop/Chatroom
tyagi@ubuntu:~$ cd ~/Desktop/Chatroom
tyagi@ubuntu:~/Desktop/Chatroom$ ./client 4444
Please enter your name: Abhishek
=== WELCOME TO THE CHATROOM ===
>
```

```
tyagi@ubuntu: ~/Desktop/Chatroom
tyagi@ubuntu:~$ cd ~/Desktop/Chatroom
              Desktop/Chatroom$ ./client 4444
                ur name: Kartik
=== WELCOME TO THE CHATROOM ===
> Abhishek has joined
>
```

Diffie Hellman Key Exchange

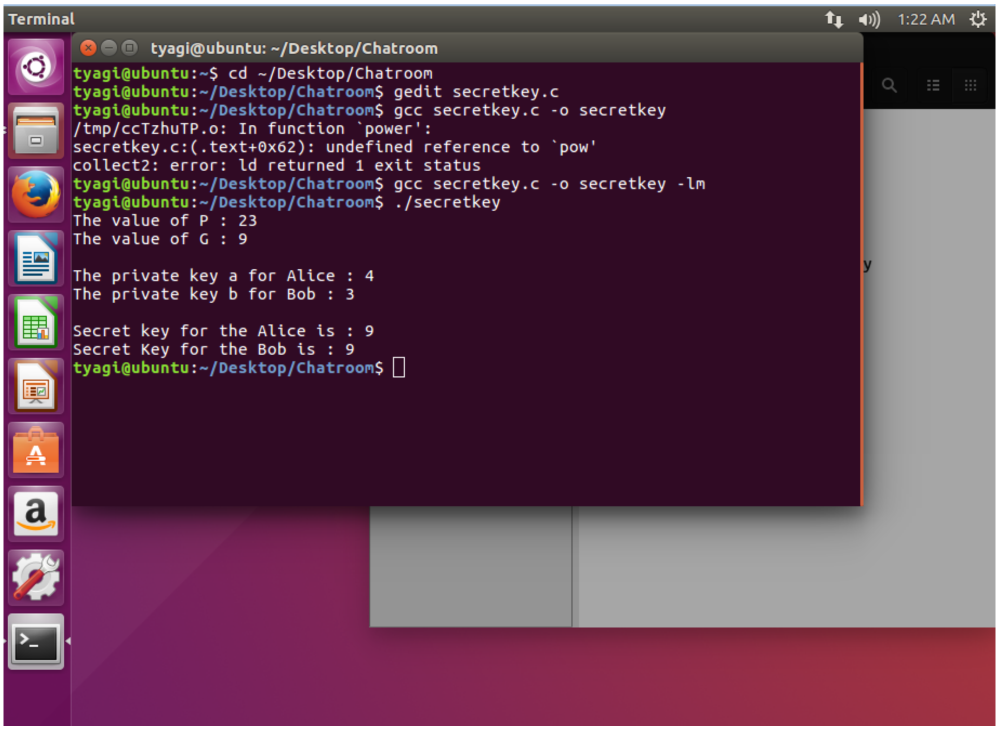
Diffie-Hellman is a public key algorithm used for producing a shared secret key. It is described in RFC 2631 and Public Key Cryptography Standard. To share a secret key between two parties, both parties calculate the shared secret key using their own private key and the other party's public key.

Algorithm

- Accept public numbers P and G
- Private key for user1: a

Private key for user2: b

- Compute public values using
  G^private key mod P
- Public key for user1: x
  Public key for user2: y
- Symmetric key(Secret key) creation from both users using
  Private key ^ Public key mod P



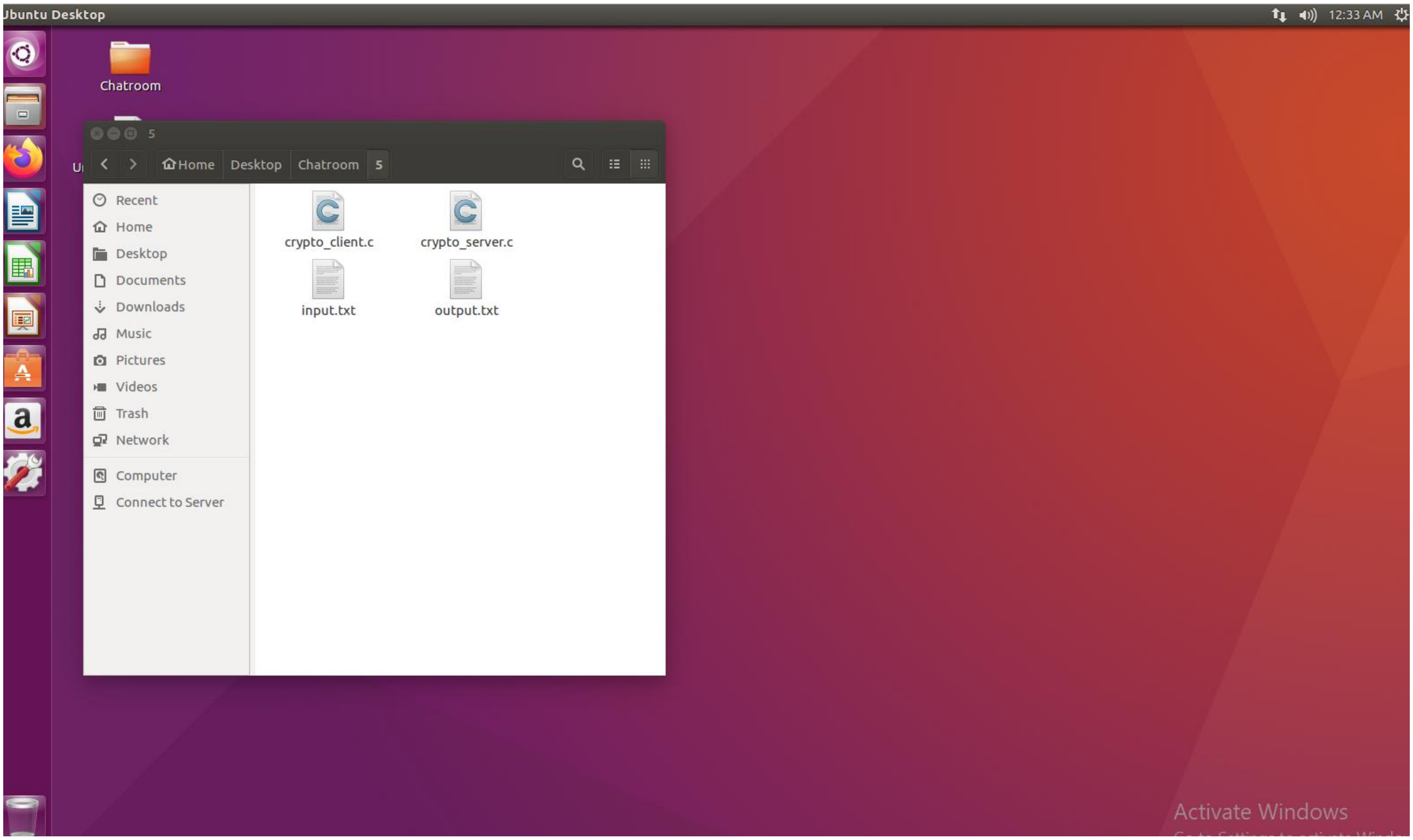
```
tyagi@ubuntu:~$ cd ~/Desktop/Chatroom
tyagi@ubuntu:~/Desktop/Chatroom$ gedit secretkey.c
tyagi@ubuntu:~/Desktop/Chatroom$ gcc secretkey.c -o secretkey
/tmp/ccTzhuTP.o: In function `power':
secretkey.c:(.text+0x62): undefined reference to `pow'
collect2: error: ld returned 1 exit status
tyagi@ubuntu:~/Desktop/Chatroom$ gcc secretkey.c -o secretkey -lm
tyagi@ubuntu:~/Desktop/Chatroom$ ./secretkey
The value of P : 23
The value of G : 9

The private key a for Alice : 4
The private key b for Bob : 3

Secret key for the Alice is : 9
Secret Key for the Bob is : 9
tyagi@ubuntu:~/Desktop/Chatroom$ 
```

IMPLEMENTED CODE

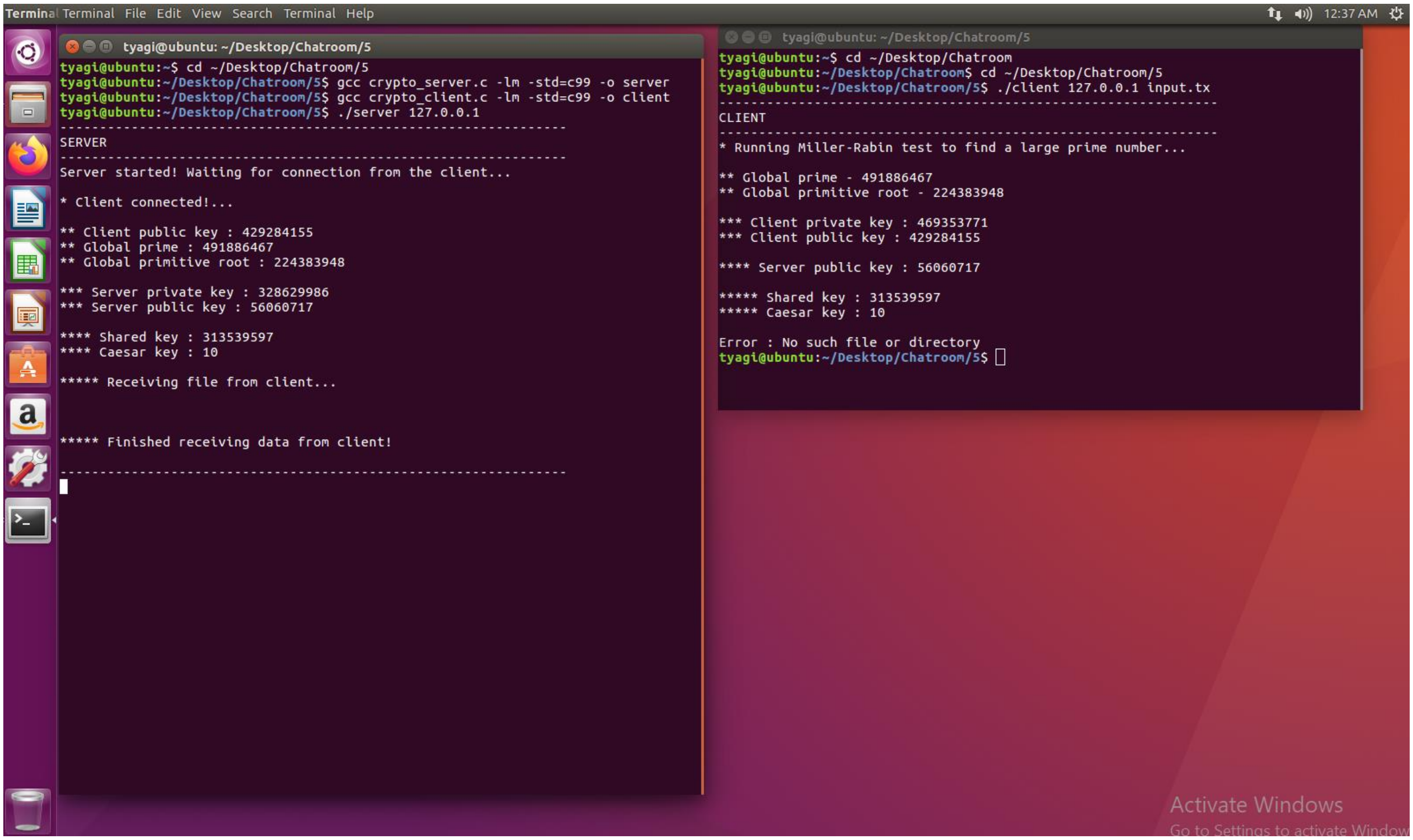
https://github.com/KartikTyagiS/Crypto-Secured-Chatbox

OUTCOMES

Source files



Running program view


Terminal Terminal File Edit View Search Terminal Help
12:37 AM
tyagi@ubuntu: ~/Desktop/Chatroom/5
tyagi@ubuntu: ~/Desktop/Chatroom/5
Activate Windows
Go to Settings to activate Windows

```
tyagi@ubuntu:~$ cd ~/Desktop/Chatroom/5
tyagi@ubuntu:~/Desktop/Chatroom/5$ gcc crypto_server.c -lm -std=c99 -o server
tyagi@ubuntu:~/Desktop/Chatroom/5$ gcc crypto_client.c -lm -std=c99 -o client
tyagi@ubuntu:~/Desktop/Chatroom/5$ ./server 127.0.0.1
----------------------------------------------------------------
SERVER
----------------------------------------------------------------
Server started! Waiting for connection from the client...

* Client connected!...

** Client public key : 429284155
** Global prime : 491886467
** Global primitive root : 224383948

*** Server private key : 328629986
*** Server public key : 56060717

**** Shared key : 313539597
**** Caesar key : 10

***** Receiving file from client...



***** Finished receiving data from client!

----------------------------------------------------------------
```

```
tyagi@ubuntu:~$ cd ~/Desktop/Chatroom
tyagi@ubuntu:~/Desktop/Chatroom$ cd ~/Desktop/Chatroom/5
tyagi@ubuntu:~/Desktop/Chatroom/5$ ./client 127.0.0.1 input.tx
----------------------------------------------------------------
CLIENT
----------------------------------------------------------------
* Running Miller-Rabin test to find a large prime number...

** Global prime - 491886467
** Global primitive root - 224383948

*** Client private key : 469353771
*** Client public key : 429284155

**** Server public key : 56060717

***** Shared key : 313539597
***** Caesar key : 10

Error : No such file or directory
tyagi@ubuntu:~/Desktop/Chatroom/5$
```

SCOPE OF PROJECT

1. The main goal of this Open Source Neural Network Library is that users can make intelligent C applications with the help of some basic training and modeling functions of this library without any prior knowledge of Artificial Intelligence. 2. This library can train very large ANN models faster than many existing libraries due to implementation of CPU and GPU optimization. 3. This is the only Open Source Neural Network Library to implement a fully generic neural network in 100% C language with CPU , GPU optimization and all the preprocessing tools within it.

SYSTEM REQUIREMENTS

Hardware:

- RAM: 2GB
- Disk Space: 4GB

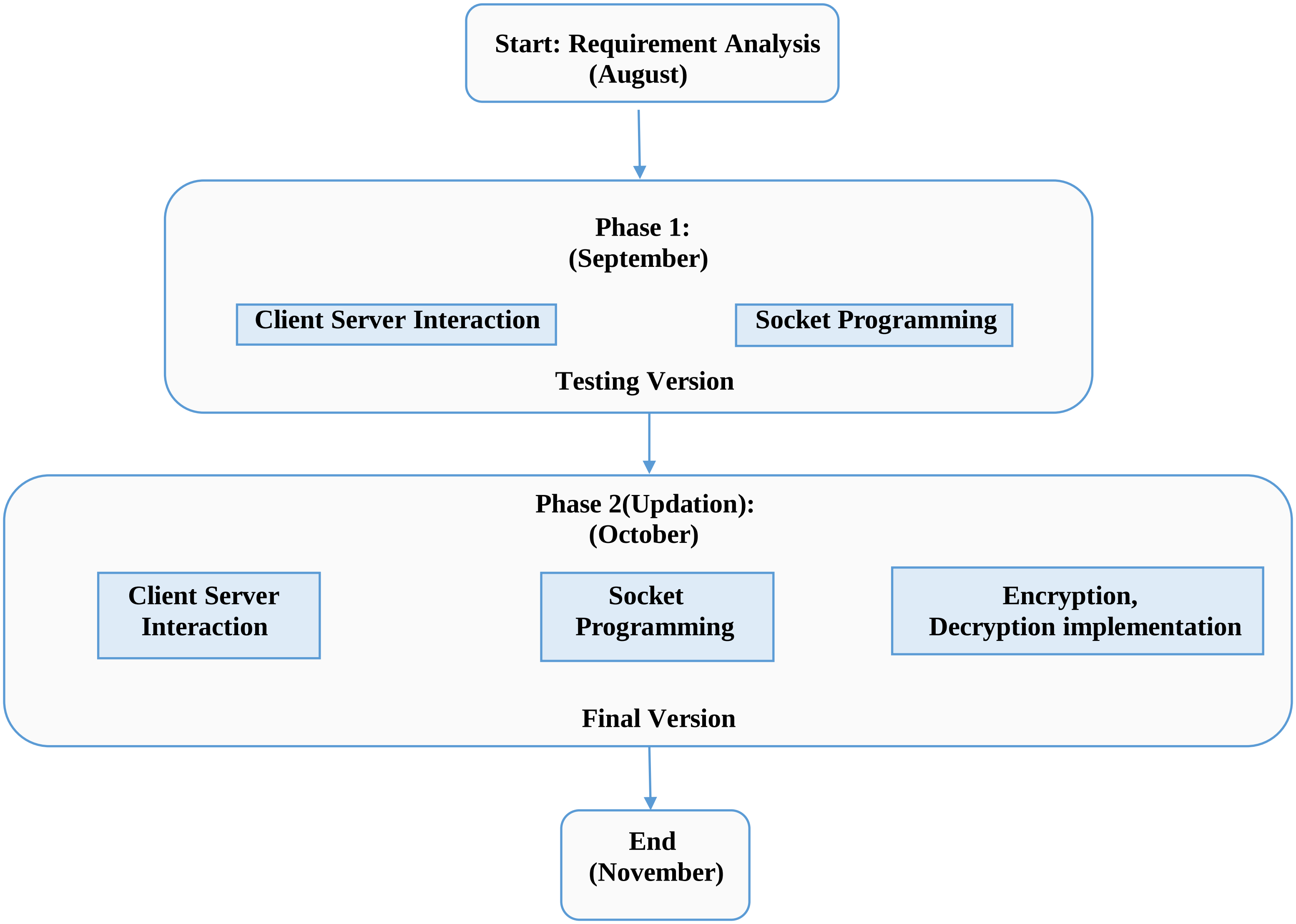
Software:

- C IDE or Dev C++ or Ubuntu

Operating System:

- Linux

PERT CHART



CONTRIBUTION OF EACH MEMBER

Abhishek Joshi
Contribution in coding(Phase 1) and major contribution in presenting the project.
Kartik Tyagi
Contribution in coding(Phase 2) and documenting the project.
Sahil Sangwan
Contribution in coding(both phase 1 and 2).

REFERENCES

[1]**https://economictimes.indiatimes.com/small-biz/security-tech/security/zomato-hacked-security-breach-results-in-17-million-user-data-stolen/articleshow/58729251.cms**
[2] **https://www.geeksforgeeks.org/**
[3] **https://en.wikipedia.org/wiki/RSA_(cryptosystem)**
[4]https://app.lucidchart.com/documents/edit/02d37549-7141-4bfd-b633-f98a6780a72e/0_0(DFD Creation tool)
[5]https://www.google.com/search?q=customer+food+booking+dfd&rlz=1C1GCEA_enIN883IN883&sxsrf=ALeKk00qqOq30-c8VnHTJ3M3N7cgj0_Qdg:1598173767228&source=lnms&tbm=isch&sa=X&ved=2ahUKEwje67yd_bDrAhVS4HMBHUZHCLYQ_AUoAXoECA4QAw&biw=1536&bih=722 (DFD examples)

Approved By

Mr. Sandeep Kumar Chaurasiya	**Dr. Monit Kapoor**
Department of Cybernetics	**Department of Cybernetics**
Project Guide	**Head of Department**