KANTIPUR ENGINEERING COLLEGE

(Affiliated to Tribhuvan University)

Dhapakhel, Lalitpur

[Subject Code: CT755]

A MAJOR PROJECT REPORT ON

END-TO-END ENCRYPTION CHATAPP

Submitted by:

Anup chaudhary [KAN075BCT009]

Chris Gurung [KAN075BCT023]

Himalaya Pal [KAN075BCT029]

Kundan Giri [KAN075BCT030]

A MAJOR PROJECT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE DEGREE OF BACHELOR IN COMPUTER ENGINEERING

Submitted to:

Department of Computer and Electronics Engineering

March, 2023

END-TO-END ENCRYPTION CHATAPP

Submitted by:

Anup chaudhary [KAN075BCT009]

Chris Gurung [KAN075BCT023]

Himalaya Pal [KAN075BCT029]

Kundan Giri [KAN075BCT030]

A MAJOR PROJECT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE DEGREE OF BACHELOR IN COMPUTER ENGINEERING

Submitted to:

Department of Computer and Electronics Engineering
Kantipur Engineering College
Dhapakhel, Lalitpur

March, 2023

COPYRIGHT

The author has agreed that the library, Kantipur Engineering Collage, may make this report freely available for inspection. Moreover the author has agreed that permission for extensive copying of this report for scholarly purpose may be granted by the supervisor(s), who supervised the project work recorded herein or, in their absence, by the Head of the Department wherein this project was done. It is understood that due recognition will be given to the author of this report and to the Department of Computer and Electronics Engineering, Kantipur Engineering College in any use of the material of this report. Copying or publication or other use of this report for financial gain without approval of the Department of Computer and Electronics Engineering, Kantipur Engineering College and author's written permission is prohibited.

Request for permission to copy or to make any other use of the material in this report in whole or in part should be addressed to:

Head
Department of Computer and Electronics Engineering
Kantipur Engineering College
Dhapakhel, Lalitpur
Nepal

KANTIPUR ENGINEERING COLLEGE
DEPARTMENT OF COMPUTER AND ELECTRONICS ENGINEERING

APPROVAL LETTER

The undersigned certify that they have read and recommended to the Institute of Engineering for acceptance, a project report entitled ”End-To-End Encryption Chatapp ” submitted by

Anup chaudhary	[KAN075BCT009]
Chris Gurung	[KAN075BCT023]
Himalaya Pal	[KAN075BCT029]
Kundan Giri	[KAN075BCT030]

in partial fulfillment for the degree of Bachelor in Computer Engineering.

....
Supervisor
none
Supervisor’s Designation
Second Line of Designation (if required)

....
External Examiner
External’s Name
External’s Designation
Second Line of Designation (if required)

....
Er. Rabindra Khati
Head of Department
Department of Computer and Electronics Engineering

Date: March 29, 2023

ABSTRACT

Data security is a crucial concern that ought to be managed to help protect vital data. Cryptography is one of the conventional approaches for securing data and is generally considered a fundamental data security component that provides privacy, integrity, confidentiality, and authentication.

In the current world where communication has been made easy such that you could talk to a person on the other side of the world with a press of the button. With the increase in availability of internet service you can send texts, photos ,files through the internet in a matter of seconds and for far less cheaper. This is achieved through different chat applications. With the increased usage of such chat applications the contents of such messages contains more that just simple messages to friends and families but also very important information and files which on the wrong hands could cause a huge catastrophe. As such End-to-End security is needed to safely exchange private information with each other without worrying about data. With this project we aim to provide an End-to-End encrypted chat apps.

This project approach End-to-End encryption method to provide secure communication that prevents third parties from accessing data while it's transferred from one end system or device to another. The End-to-End encryption method is implemented using the Asymmetric key encryption algorithm (RSA). We used RSA to encrypt that encoded data. In this way we can maintain the data more securely.

Keywords– RSA

ACKNOWLEDGMENT

We would like to acknowledge Er. Bikash Shrestha, our project supervisor who is providing us guidence and helping us with different views of application, development and scalability as well as helped us in development of our project. His guidence has been corner stone in developing chat app. We are able to accomplish this project with his valuable direction and suggestions.

We would like to express our notable thanks to Department of Computer and Electronics Engineering for providing us environment to dig the latest technology, investigate it and do research on those areas as Major project. We are also grateful to Er. Rabindra Khati, HOD, Department of Computer and Electronics Engineering for presenting useful suggestions regarding the queries put forward regarding the project.

Our special appreciation goes to Lecturers of Kantipur Engineering College for guiding us in this project and helping to better cope with the difficulties along with giving supportive hands. Without their valuable guidence it would have been difficult for us to come to a conclusion to work with this technology.

Last but not least, we would appreciate all our seniors and friends for their support and constant inspiration which helped us to withstand in our success in this project.

Anup chaudhary	[KAN075BCT009]
Chris Gurung	[KAN075BCT023]
Himalaya Pal	[KAN075BCT029]
Kundan Giri	[KAN075BCT030]

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

1.1 Background

With the rapid development of mobile devices, computers and accessibility to the internet there is growing users of different chat applications. The attracting or more so important features of any social media has become the chat feature. Such chat features provide real time messaging, file sharing which may include different photos, videos, documents, etc. Chatting has been such an important aspect of the modern world that it is expected the no. of active users currently is in billions with this number expected to increase more in the coming years. Among some of the popular chat applications Whatsapp's monthly active users(in millions) are 2000, messenger 988, snapchat 557 to name a few.[1]

The importance of such apps was never more clearly presented as when Facebook experienced outage in 2021. In what was just a 6 hour long outage the competing apps like telegram gained a record 70 million new users. As the dependence on chat systems grow day by day the vulnerability and assaults also increases. As such there is increasing need to implement a secure communication

Digital communication witnesses a noticeable and continuous development in many applications in the Internet. Hence, secure communication sessions must be provided. The security of data transmitted across a global network has turned into a key factor on the network performance measures. So, the confidentiality and the integrity of data are needed to prevent eavesdroppers from accessing and using transmitted data.

One way of to secure the communication is using End-To-End encryption which use the Cryptographic algorithm. Cryptographic algorithms are divided into symmetric and asymmetric keys. The symmetric algorithms require a single key only for the encryption and decryption of data. Asymmetric algorithms on the other hand require both public and private keys for the encryption and decryption of data. The scrambling of the data is done using the public key while the private key is made known only to the receiver which is meant for the decryption of the data. A maiden asymmetric algorithm was proposed by Diffie-Hellman which ensures secured communication as well as data security. A counter algorithm termed RSA which has lower time complexity based on prime number factorization was proposed in 1977 and is the patent of Ron Rivest, Adi Shamir, and Len Adleman (RSA), which was published in 1978 at the Massachusetts Institute of Technology . In this algorithm, two prime numbers are used to produce the public and the private key. When the keys are created, the prime numbers are no more considered and are or can be discarded.[2]

1.2 Problem statement

Traditional messaging like SMS are very unsecure as it travels through the network in plain text. Since messages might contain very sensitive informations the messages shouldn't be accessible to any other person beside the actual people involved in the conversation. Furthermore in unsecure messaging any third party can pretend to be the intended person and gain access or send wrong information.

1.3 Objectives

The application aims to provide data security in communication system. The application focuses on simplicity of design, having user-friendly interface and to be easily understood.

The main objectives of the application can be enumerated as follows:

- To provide end-to-end text message security.
- To provide platform for sending messages from one person to another.

1.4 Applications

The application is an online web application. This system can be used to provide end to end encryption of data and also used to provide secure connection between user between users with smooth and clean UI.

1.5 Project features

The application, is targeted towards the general population, so the core features of this applications can be listed below:

- We can send messages
- It provides end-to-end data security in communication system.
- Simple and ease for general population

1.6 Feasibility Analysis

Feasibility is a measure of how benificial or practical the development of a system will be to an organization or enterprise. In other words, a feasibility study is an evaluation and analysis of the potential of the proposed project which is based on extensive investigation and reasearch to give full comfort to the decision masker. A system should have a responsible cost and should be technically and operationally feasible to be actually implemented. The two criteria to judge feasibilty are cost required and value to be attained. The feasibility of ”End to End encryption Chatapp ” is analyzed under the following four headings:

1.6.1 Economic Feasibility

Economic Feasibility is the measure of the cost effectiveness of the project or the solution. This is often called a cost benifit analysis.

Based on our economic analysis for development and operational cost, the system is being developed and operated economically. For development, the required devices are readily available, so it is feasible. Also, it is economically feasible to the consumers as it costs no charge to use the platform.

1.6.2 Schedule Feasibility

Based on the objectives and the time left for the development. The schedule is found to be feasible.

1.6.3 Technical Feasibility

Technical Feasibility is the measure of the practically of a specific technical solution and the availability of technical resources and expertise.

Technically, the system is feasible enough and easy-to-use for both technical and non-technical groups of people. It provides a user-friendly environment along with features using the latest technologies. The system provides a layout most of the applications people are used to anyway so it will be easy to use.

1.6.4 Operational Feasibility

Operational feasibility is the measure of how a well specific solution will work in the organization. It is also measure of how people feel about the system of the project. It also analyze the inside operation on how a deemed process will work, be implemented, and dealing with changes resistance and acceptance.

For the operation of the system, the person does not need to excel in using a computer. Since, the event may not always be related to the technical fields, someone with minimum knowledge about computer and technology can also get benefit from the system. Similarly, one can get access to the system as a web-based application. There is no requirement of huge and expensive hardware.

1.7 System Requirements

1.7.1 Software Requirement

Application is targeted towards a general market, so it is aimed to be fully optimized enough for any low-range to high-range systems, so listed below are the software requirements for the development and operation of this system:

- Operating System: Windows 8 or above
- Browsers: Google Chrome, Firefox, etc
- Mongodb, Node.js Server
- React js

1.7.2 Hardware Requirement

Hardware configuration and requirements for the operation of this application are as follows:

- Intel Core 2 Duo Processor (Recommended i-series processors or more) with minimum of 2GB RAM for application operation

- Server with optimum node speed

CHAPTER 2
LITERATURE REVIEW

2.1 Related Paper

There are currently millions of monthly active users worldwide of different chat applications currently. 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.[3]

There are different Encryption algorithms that can be utilized to provide secure messaging environment. Thus, messages will circulate as encrypted form in transmission medium, not as clear text. Somebody who has seized encrypted data does not obtain original message from the encrypted data unless they possess the necessary method or a key. Encryption methods are divided into the following categories: private key cryptography and public key cryptography.

In a symmetric key algorithm, the sender and receiver must have a shared key set up in advance and keep secret from all other parties; the sender uses this key for encryption, and the receiver uses the same key for decryption. In this case, except for transmitted encrypted message, encryption key must also be submitted confidentially, which is one of the disadvantages of private-key cryptography. If a third person who has managed to enter the system operator or listen to transmission medium seizes the key value, s/he can turn the encrypted data into original data. The most important feature of public key cryptography which is another method is that the key value used to encrypt the message is different from the key value used to decrypt the message. Each user has two keys in this method: public key and private key. The public key of the user can be viewed by anyone. The private key is kept secret by the user. When someone wants to send a message to user, they use the user's public key and create the encrypted message and then send the encrypted data to user. The user decrypts the encrypted data with her/his private key and obtains a meaningful message.

The data which has been encrypted by the user's public key is only solved with the user's private key. When the user wants to send a message, s/he reaches the public key library. S/he takes the public key of somebody to which s/he wants to send a message and encrypts the message and then s/he sends the encrypted message to the receiver. The only thing that the receiver must do is to solve the message with his/her own private key. DH protocol is to enable two users to exchange a secret key securely that can be used for subsequent encryption of messages. The protocol itself is limited to the exchange of the keys. But because of having no entity authentication mechanism, DH protocol is easily attacked by the man-in-the-middle attack and impersonation attack in practice. The DH key exchange is a cryptographic protocol that allows two parties that have no prior knowledge of each other to establish together a shared secret key over an insecure communications channel. Then they use this key to encrypt subsequent communications using a symmetric-key cipher. [4]

The Classic Diffie-Hellman (CDH) algorithm encounters several security problems. The algorithm security depends on the complexity of solving the discrete logarithm and the integer factorization problem. The security also depends on the length of bits keys used. In this paper, we proposed a hybrid algorithm, the combination of Modified Diffie Hellman (MDH) with the Rivest, Shamir, and Adelman (RSA) algorithm. The MDH will not use primitive root (g). Instead, it will use two prime numbers (P and Q) for key exchange. It reduces the possibility of a man-in-the-middle attack. Furthermore, the MDH has an authentication mechanism. If both shared keys are equal, the communication will continue. Otherwise, it will stop because it is compromised. Also, the RSA algorithm will use the two prime numbers generated by MDH (P and Q). RSA algorithm, which is one of the public-key encryption methods and more reliable than

the private key encryption algorithms, is used for secure messaging.[5]
RSA algorithm is asymmetric cryptography algorithm. The RSA algorithm will perform the encryption and decryption message. The Diffie Hellman (DH) and the RSA Algorithms are the basis of several security standards and services on the internet, especially TLS and SSL. If the security of both algorithms is compromised, such systems will collapse. In this proposal, the combined cryptography system aims to achieve secret message exchange; it uses random prime numbers after multiplication and eventually shared with the other end of a communication. The proposed algorithm secretly generate a value so that the security level increases .[4]
Security of RSA depends on the prime factorization of N. The proposed algorithm authenticates a user as shown is the first part of the algorithm by exchange value of random P and Q. Another strength of the hybrid algorithm, it will not generate primitive root (g). Then generate secret keys at both sides. Secret keys not known or not sent over a communication channel, so eavesdropper has no chance to get the value of the secret key. The secret key is multiplied to P, and hence new P will be not identifiable to attacker though the value of P gets compromised. This value of P and Q is generated secretly at the user side, and this valued is not shared through the communication channel, so we claim that the proposed method is more secure than the original RSA. The disadvantage of the proposed algorithm has extra steps that mix symmetric key and asymmetric key cryptography. The time required for this integration steps is disadvantageous of this proposed method. But we can achieve better security[4]

2.2 Existing system:

In this section we briefly introduce many of the popular chat applications. Some of these applications are not public or open source so it is difficult for these to get evaluated by the developer's community, security experts or research academic.

2.2.1 Viber

Viber is an instant messaging and Voice over IP (VoIP) application for smartphones developed by Viber Media. In addition to instant messaging, users can exchange images, video and audio media messages. Viber recently supported the end-to-end encryption to their service, but only for one-to-one and group conversations in which all participants are using the latest Viber version 6.0 for Android, iOS or Windows 10. At this time, in the Viber iOS application for iPhone and iPad, attachments such as images and videos which are sent via the iOS Share Extension does not support end-to-end encryption .[6] Viber has privacy issues such as adding a friend without his knowledge or adding him to a group without his permission. Plus that, local storage is not secured. It is not open source making it difficult to evaluation.

2.2.2 Whatsapp

WhatsApp is one of the most popular messaging application, recently enabled end-to-end encryption for its 1 billion users across all platforms. WhatsApp uses part of a security protocol developed by Open Whisper System, so provides a security-verification code that can share with a contact to ensure that the conversation is encrypted. It is difficult to trust in WhatsApp application completely because the application is not open source, making it difficult to verify the functioning process and match them with the work of the encryption protocol which was announced. .[7]

2.2.3 Telegram

Telegram is an open source instant messaging service enables users to send messages, photos, videos, stickers and files. Telegram provides two modes of messaging is regular chat and secret chat. Regular chat is client-server based on cloud-based messaging, it does not provide end-to-end encryption, stores all messages on its servers and synchronizes with all user devices. More, local storage is not encrypted by default. Secret chat is client-client provides end-to-end encryption. Contrary to regular chat messages, messages that are sent in a secret chat can only be accessed on the device that has been initiated a secret chat and the device that has been accepted a secret chat they cannot be accessed on other devices. Messages sent within secret chats can be deleted.[8] at any time and can optionally self-destruct. Telegram uses its own cryptographic protocol MTProto which is based on 256-bit symmetric AES encryption, 2048-bit RSA encryption, and Diffie–Hellman secure key exchange, and has been criticized by a significant part of the cryptographic community about its security. The registration process of Telegram, Viber and WhatsApp depend on SMS. SMS is transported via Signaling System 7 (SS7) protocol. The vulnerability lies in SS7. Attackers exploited SS7 protocol to login into victim's account by intercepting SMS messages. Because of Telegram cloud-based, the attacker exploits it and makes full control of the victim account and can prevent him to enter into his account. To make the account more secure should activate two-factor authentication [9]

2.2.4 Facebook Messenger

Facebook Messenger is a popular messaging service available for Android and iOS. It provides two modes of messaging one is regular chat and another is secret conversations. Regular chat does not provide end-to-end encryption only secure communication by using TLS, and it stores all messages on its servers. Secret conversations have the same idea of Telegram secret chat . .[10]

2.3 The Challenge- Making it real-time

For any app to feel real-time, the user needs to be kept updated with any activity happening as soon as possible. The challenge arises in selecting and implementing a suitable development technique. With the traditional request-response model, we have few options:

2.3.1 Refresh Webpage

The user might refresh the web page time-to-time to check for message updates. But that is not an optimal solution. This may result in bad UX.

2.3.2 HTTP Protocol

The concept of HTTP request-response is widely used. But this requires establishing a TCP connection every time data is sent to the server. Being a one-way synchronous communication protocol, this may result in a lot of overheads while creating and destroying a TCP connection every time a message is sent in real-time chat applications.

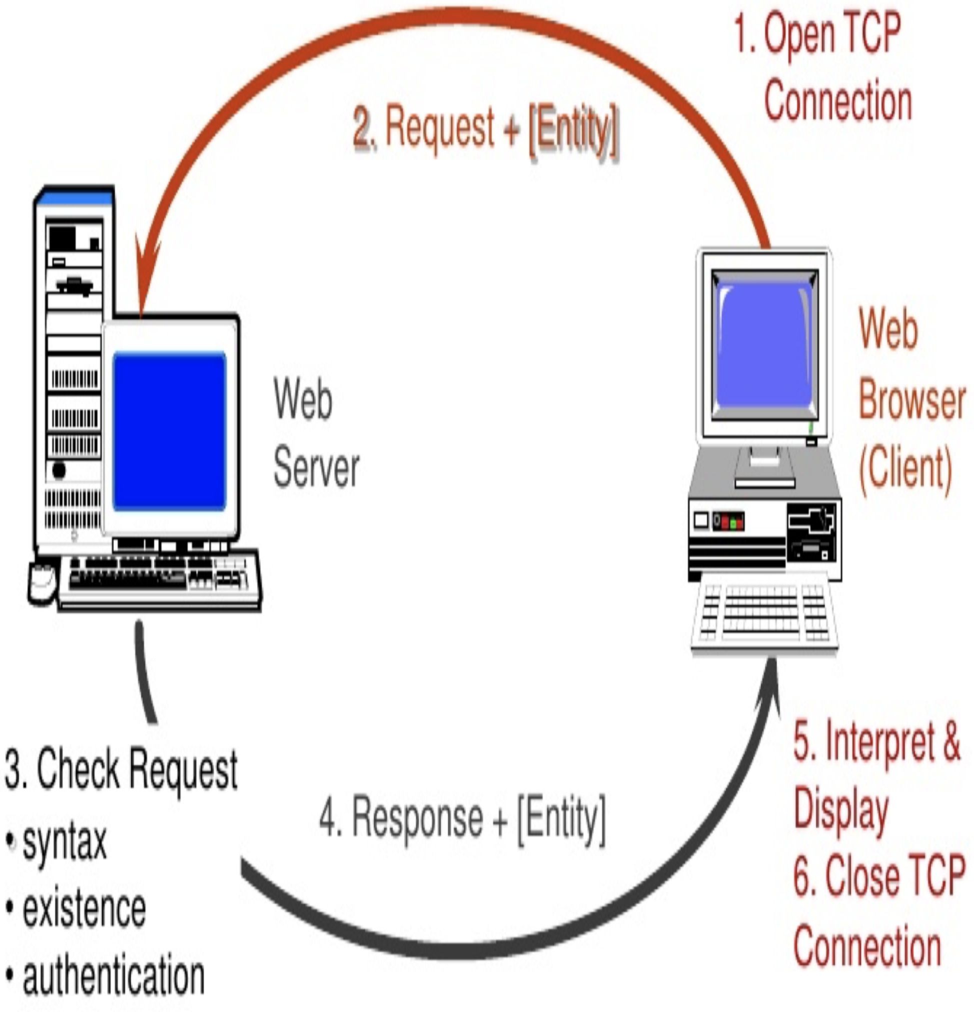


Figure 2.1: Http Request-Response Cycle

2.4 Web Sockets

This concept resolves most of the issues we just discussed. It implements instant two-way communication of messages with a persistent connection just as required for developing a real-time system. NodeJS offers several libraries to implement this technology. What we will be utilizing for our application is the Web Socket API with 'Socket.io' library.

2.4.1 Socket.io Api

Socket.IO is a library that enables low-latency, bidirectional and event-based communication between a client and a server. It is built on top of the WebSocket protocol and provides additional guarantees like fallback to HTTP long-polling or automatic reconnection.

2.4.2 Socket.io Handshake

The handshake in Socket.IO is like any other information technology related handshake. It is the process of negotiation, which in Socket. IO's case, decides whether a client may connect, and if not, denies the connection.

2.4.3 Exchanging Messages

- The messages are exchanged in the form of data frames rather than a stream of data
- It is a bi-directional flow of data
- The event listeners of the ws object are used for message exchange
- The closing handshake can take place either by the client or the server. Reconnection has to be done manually.

CHAPTER 3
METHODOLOGY

3.1 Required Algorithm:

3.1.1 Overview of Diffie Hellman Algorithm

The algorithm is based on Elliptic Curve Cryptography, a method of doing public-key cryptography based on the algebra structure of elliptic curves over finite fields. The DH also uses the trapdoor function, just like many other ways to do public-key cryptography. The simple idea of understanding to the DH Algorithm is the following.

3.1.2 Diffie-Helmann Algorithm

With simple encryption, the messages are usually encrypted only between the users and the server, making use of some cryptographic keys, hence making data vulnerable at the server. We want these keys only to exist between the users and not the server. Diffie-Helmann makes this possible. Suppose we have two clients ALice and Bob.

- Alice and Bob agree to use two common prime numbers (g & n) provided by the server
- Now, these are combined using some mathematical calculations with the Private keys of Alice and Bob = a + g = ag and b + g = bg.
- We exchange these Ephemeral/Public Keys ag and bg via server.
- Combine the exchanged keys with the Private keys of Alice and Bob respectively to form a Shared Secret Key = ag+b = agb and bg+a = bga at both ends.
- Now the attacker might be aware of g, n, ag & bg as these are being shared publicly, but not a & b since these are private keys only available to Alice and Bob.
- It is too difficult for any intruder to split up the public components ag and bg.
- Any attacker can combine ag+bg = abgg (extra bit) - too hard to figure out.

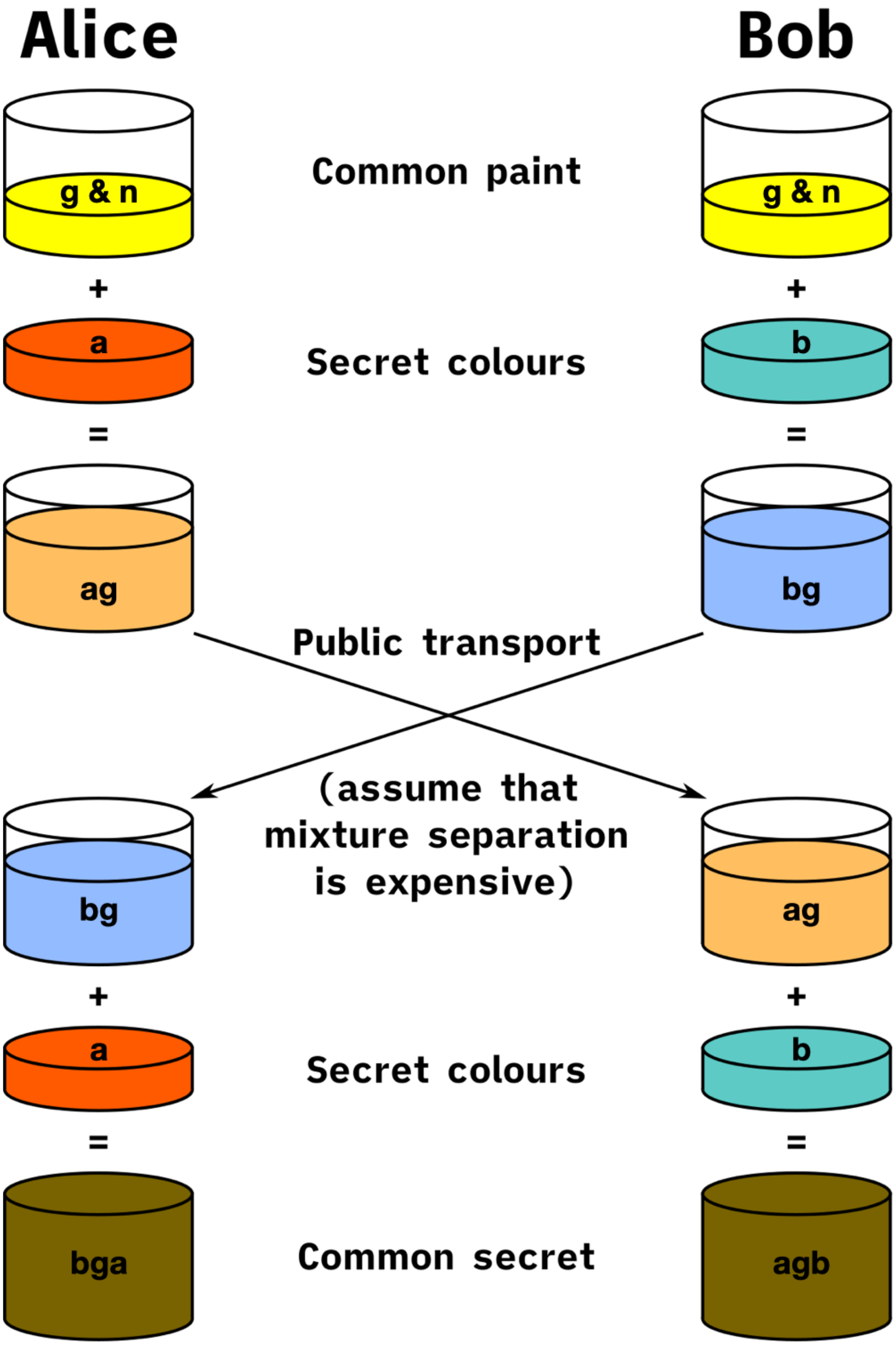


Figure 3.1: Diffie Hellman working mechanism

This mechanism was developed by Whitfield Diffie and Martin Hellman to derive the cryptographic keys instead of exchanging them completely in public. It is explained using colors since it is not possible to separate colours once mixed. Similarly, it is hard to figure out the secret keys using the only public components, once combined mathematically with the prime numbers provided by the server.

3.1.3 Problems with Diffie-Hellman Key Exchange

Although the mechanism provides us with a secure way to create cryptographic keys as end-to-end, it does not authorise the users. Hence, we might have some third party pretending to be the intended recipient and he/she will be able to access or modify the messages, by creating another pair of shared secret keys with Alice and Bob respectively. This is usually known as a Man-in-the-middle attack.
That is when RSA came to rescue. The sender not only performs Diffie-Hellman but also shares his/her signature to ensure that only he/she has sent that message.

3.1.4 Proposed algorithm

Steps:

- Alice(User A) and Bob (User B) Choose two very large random prime integers: s and r
- Calculate n = s*r and z = (s-1)(r-1)
- Choose a small exponent say e :

  But e Must be

  An integer.

  Not be a factor of n.

  1 < e < z
- Calculate d = e^{-1} mod(p-1)(q-1) or d = (k*z + 1) / e
- Public key will contain (n , e) and private key will contain (n , d).
- Alice needs to identify Bob. For the authentication process Alice and Bob use Modified Diffie-Hellman algorithm
- Alice and Bob choose two large prime p,g such as g<p.
- Alice chooses a large random number x$_1$$(0 < x_1 < p)$ and computes $R_1 = g^{x_1} modp.$
- Alice sends R_1 to Bob
- Bob chooses a large random number x$_2$$(0 < x_2 < p)$ and computes $R_2 = g^{x_2} modp.$
- Bob computes $K_{Bob} = R_1^{x_2} modp$ and $E_1 = Encrypt(R_2, K_{Bob}).$

- Bob sends R_2, E_1 to Alice.
- Alice computes $K_{Alice} = R_2^{x_1} modp$ and $R_2^{'} = Decrypt(E_1, K_{Alice})$.
- If $R_2 = R_2^{'}$ she proceeds; otherwise the verifier is dishonest.
- Alice lets Bob send his public key.
- Bob encrypts his public key via the secret key (K_{Bob}) and sends it to Alice.
- Alice decrypts Bob's public key via the secret key (K_{Alice})
- Alice encrypt message with Bob's public key and sends it to Bob.
- Bob decrypts the message via the secret key (K_{Bob})
- Now Bob has the original message.

There are two main steps in the suggested algorithm: user authentication and encryption. These steps have been shown in figure below:

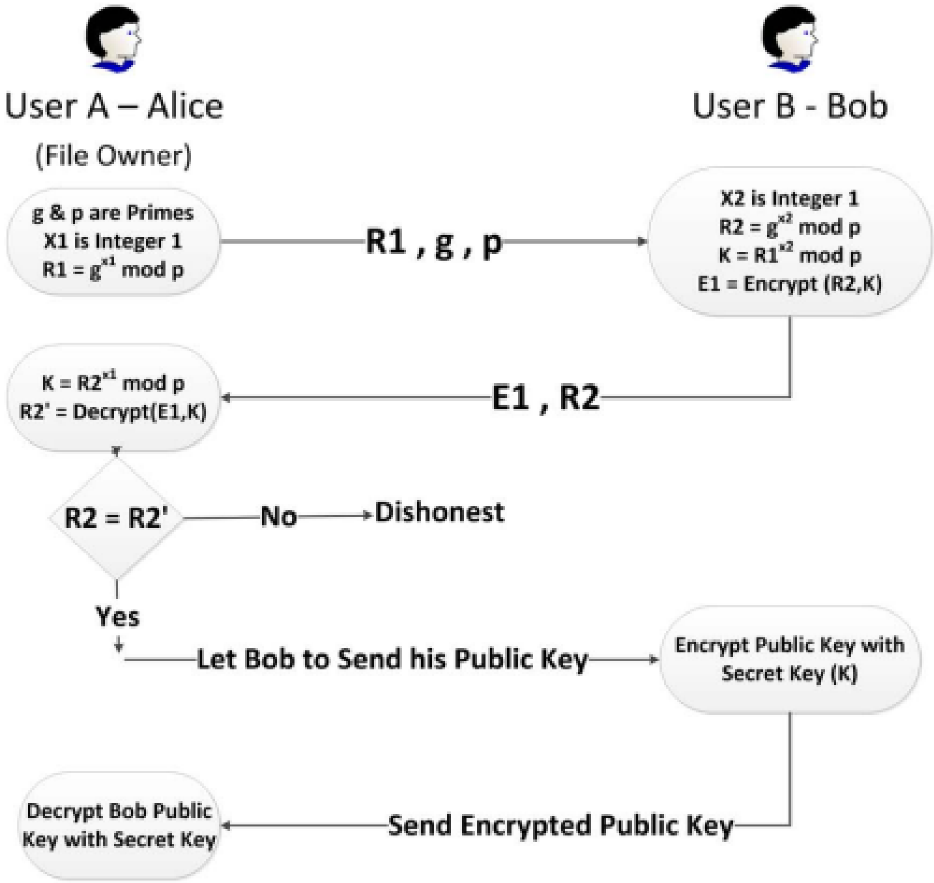


Figure 3.2: User authentication and encryption of public key

3.1.5 Overview of RSA Algorithm

RSA encryption algorithm, which is based on the idea of ensuring the secure transfer of data in the digital environment and the algorithmic difficulty of separating the integer factorization, is a type of public-key encryption method. Nowadays, it is also known as both the most commonly used encryption method and the method that allows digital signatures. It was created by Ron Rivest, Adi Shamir and Leonard Adleman in 1978. Prime numbers are used for key generation process in RSA encryption method.This makes it possible to create a safer structure. How the encryption and decryption processes are done with RSA algorithm is shown in below figure,

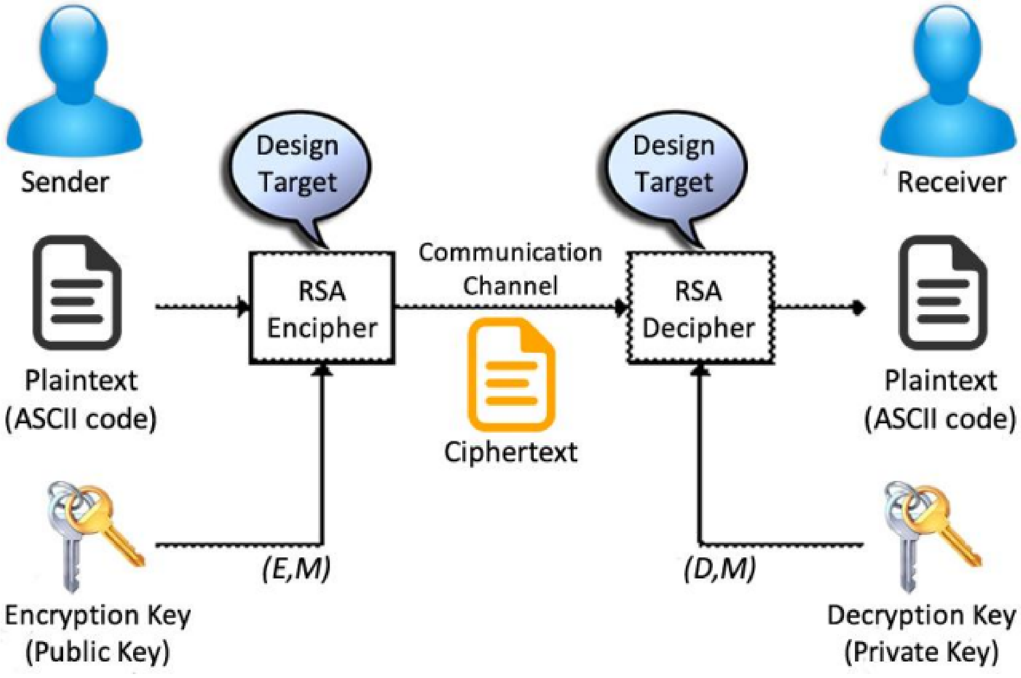


Figure 3.3: RSA algorithm working mechanism

3.1.6 RSA algorithm structure

Steps:

- Choose two very large random prime integers: p and q
- Calculate n = p*q and z = (p-1)(q-1)
- Choose a number e where $1 < e < z$
- Calculate d = e-1mod(p-1)(q-1)
- You can bundle private key pair as (n,d)
- You can bundle public key pair as (n,e)

After creating public and private keys, information which must be sent is encrypted with the public key.

Encryption and decryption processes are done as follows:

- The cypher text C is found by the equation where M is the original message
- The message M can be found from the cypher text C by the equation $M = C^d$ mod n
- A text encrypted with the public key can only be solved with the private key

3.1.7 Security analysis

Man in the Middle attack and Discrete Logarithm attacks are the most damaging attacks in key-exchanging process. Furthermore, Cycle attack, Brute Force attack, and Timing attack are the most important attacks during the encryption and decryption process. The following step evaluates the algorithm against these attacks.

Man in the Middle Attack

Man in the Middle is an attack that the attacker is able to read and modify all the messages between Alice and Bob [16]. To protect the suggested model from Man in the Middle attack, encrypted replies (R_1, R_2) and mutual authentication between Alice and Bob is required. For this purpose Bob computes $K_{Bob} = R_1^{x2} modp$ and $E_1 = Encrypt(R_2, K_{Bob})$ and sends R_2, E_1 to Alice. After that, Alice computes $K_{Alice} = R_2^{x1} modp$ and $R_2^{'} = Decrypt(E_1, K_{Alice})$. These processes prevented the Man in the Middle attack and by comparing R_2 and $R_2^{'}$ the attack will be identified. The following figure shows the Man in the Middle attack prevention in the proposed model.

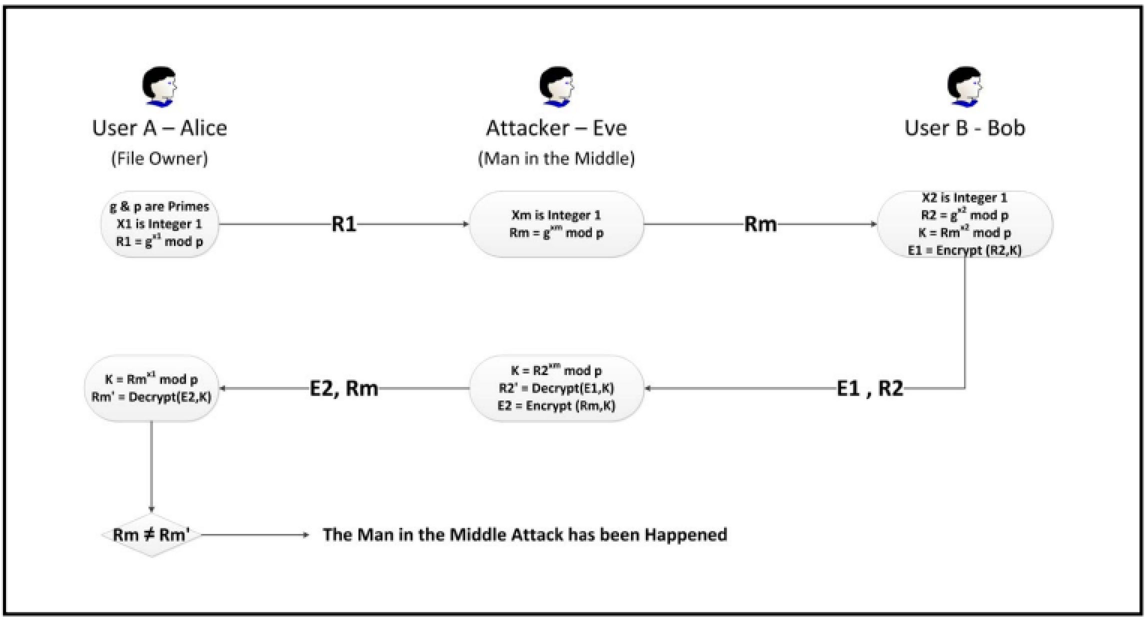


Figure 3.4: Man in the Middle Attack Prevention by Proposed Algorithm

According to the figure, R_m and $R_m^{'}$ are not same for Alice because the keys between users and attackers are different.

$K_{Alice} = g^{x_1 x_m}$ mod p

$K_{Bob} = g^{x_2 x_m}$ mod p

$K_{Eve} = g^{x_2 x_m}$ mod p

It means $K_{Alice} \neq K_{Bob}$ = K_{Eve} and because of that, the Man in the Middle attack was noticed by Alice and the attack was prevented.

Cycle Attack

Cycle Attack is an attack where the attacker encrypts the cipher text alternately, until the original text appears. This number of encrypting will decrypt any cipher text. In large key RSA, this attack could not be practical even when a generalisation of the attack allows the modulus to be factored and most of the time it works faster. Moreover, in the proposed model, the attacker will not have access to the public key to re-encrypt the cipher text because the public key has been encrypted by the secret key that was generated in Modified Diffie-Hellman process.

Brute Force Attack

All possible combinations to guess the private key have been tried by the attacker during Brute Force attack. In original RSA, the probability of failure against this attack can be decreased considerably by choosing exponents larger than 2048 bits but with the combination of the proposed model, this algorithm has significant resistance towards brute force attack even with 1024 bits exponents because of the encryption of the public key before sending it.

Timing Attack

Timing attack is a side channel attack in which the attacker determines private exponent by calculating the time by exploiting the timing variation of the modular exponentiation [18]. Timing attack in original RSA might be prevented by including a random delay to the exponentiation algorithm or multiplying the cipher-text with a random number

[19] while the dual encryption (public key encryption by secret key and the message encryption by RSA) in the suggested model will protect the transferred message from the timing attack and it is not necessary to multiply the cipher-text.

3.1.8 CIA triad

CIA triad is one of the most important models which is designed to guide policies for information security within an organization. CIA stands for :
Confidentiality, Integrity, Availability.
In this project, we only focuses on two models Confidentiality and integrity.
Confidentiality means that only authorized individuals/systems can view sensitive or classified information. The data being sent over the network should not be accessed by unauthorized individuals.We used Encryption to protect the data.

The next is integrity, data cannot be modified by third party. Corruption of data is a failure to maintain data integrity. We verified that we are sending message to a genuine person so no tampering of message will happen. To check if our data has been modified or not, we make use of a hash function.

3.2 Software development model

3.2.1 Incremental Model

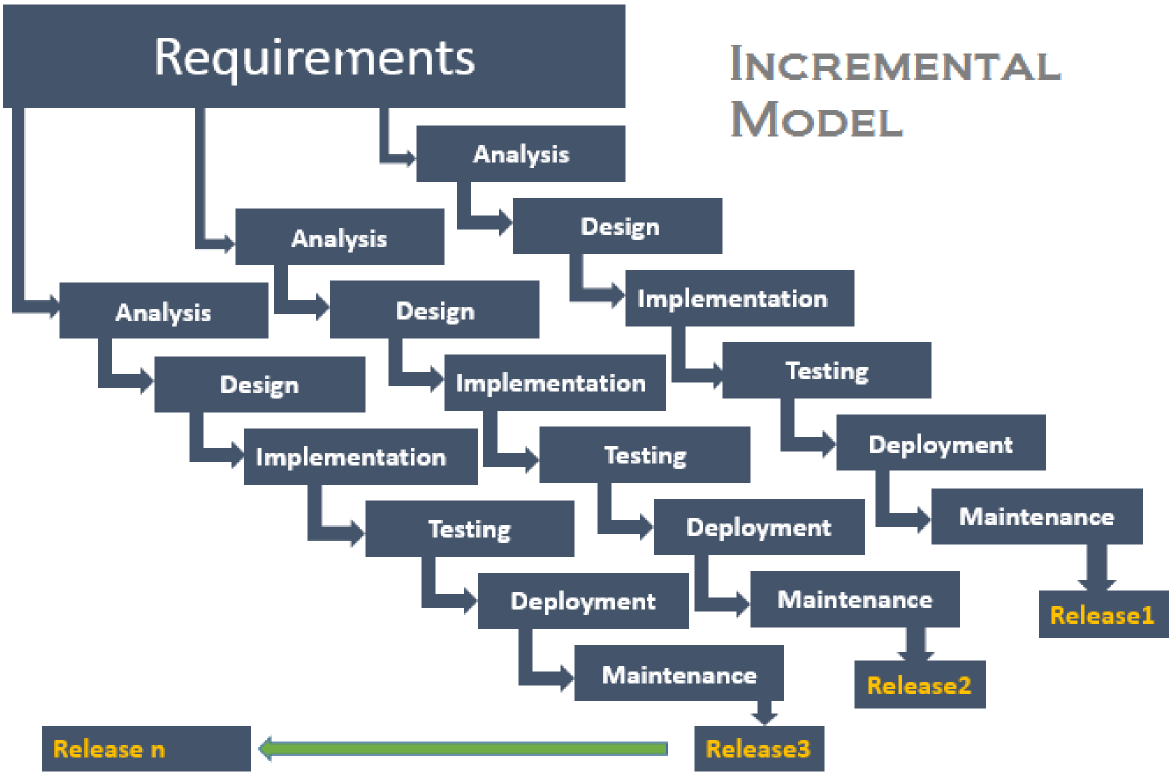


Figure 3.5: Incremental Model Block Diagram

First Increment:

First, the website was analyzed to know how it should look like. Then, the designing of templates was done. After observing the design of the templates, we started coding using react js, bootstrap, JavaScript.

Second Increment:

After analyzing the scenario of the project, the algorithms to be implemented was analyzed. Using the algorithms, we started designing the algorithms that is suitable for the project. Initiating the coding we completed the algorithm implementation. Finally, the algorithm was implemented.

Third Increment:

After the algorithm implementation backend designing and coding was started. Using Django and Python the backend part was completed and tested. Finally, backend was ready.

Fourth Increment:

Finally, after all the designing was completed, coding for the project was done. Later, testing of the project was done. At last, the final webapp was designed.

3.3 Block Diagram:

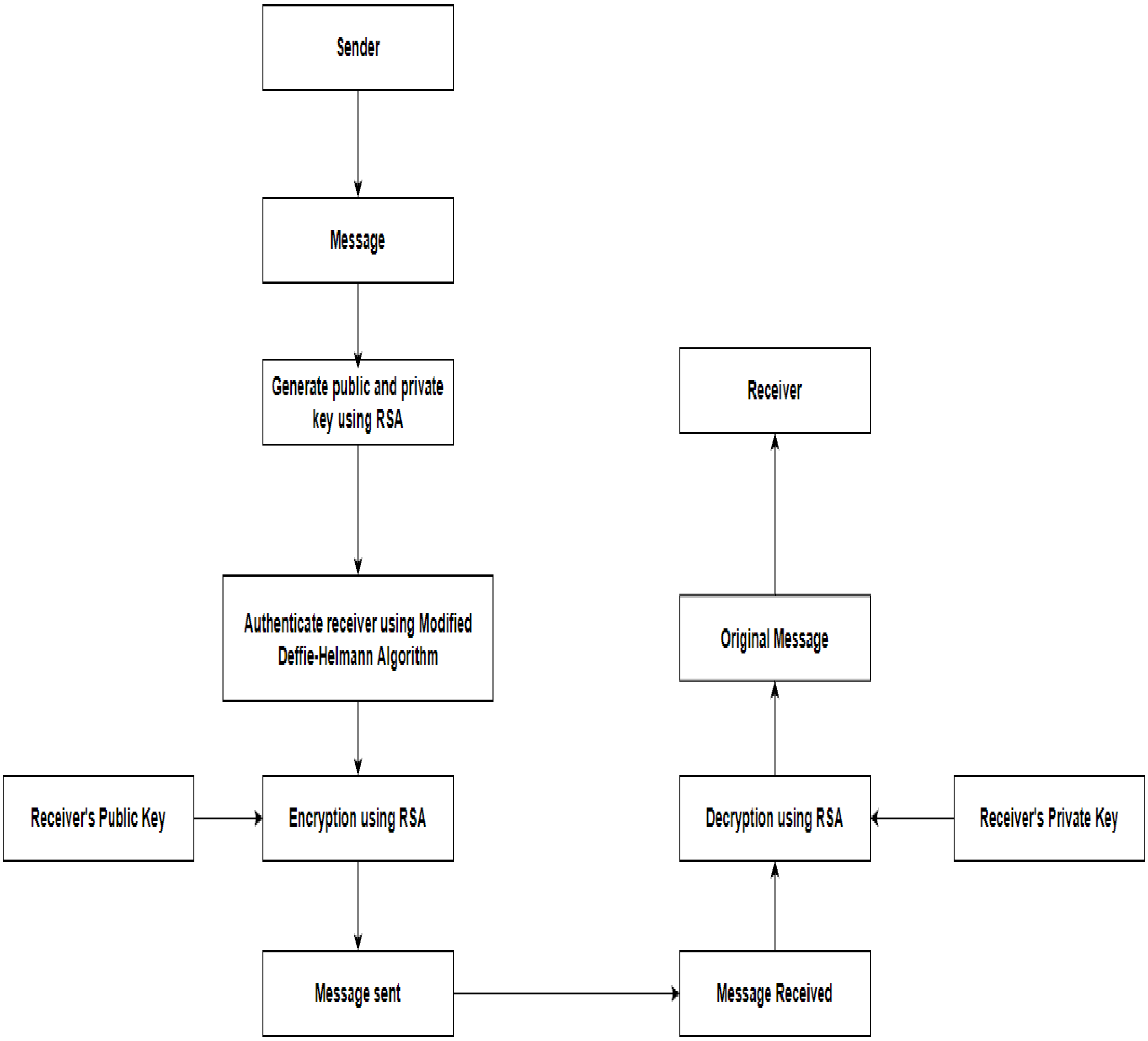


Figure 3.6: Block Diagram

3.4 Use Case Diagram:

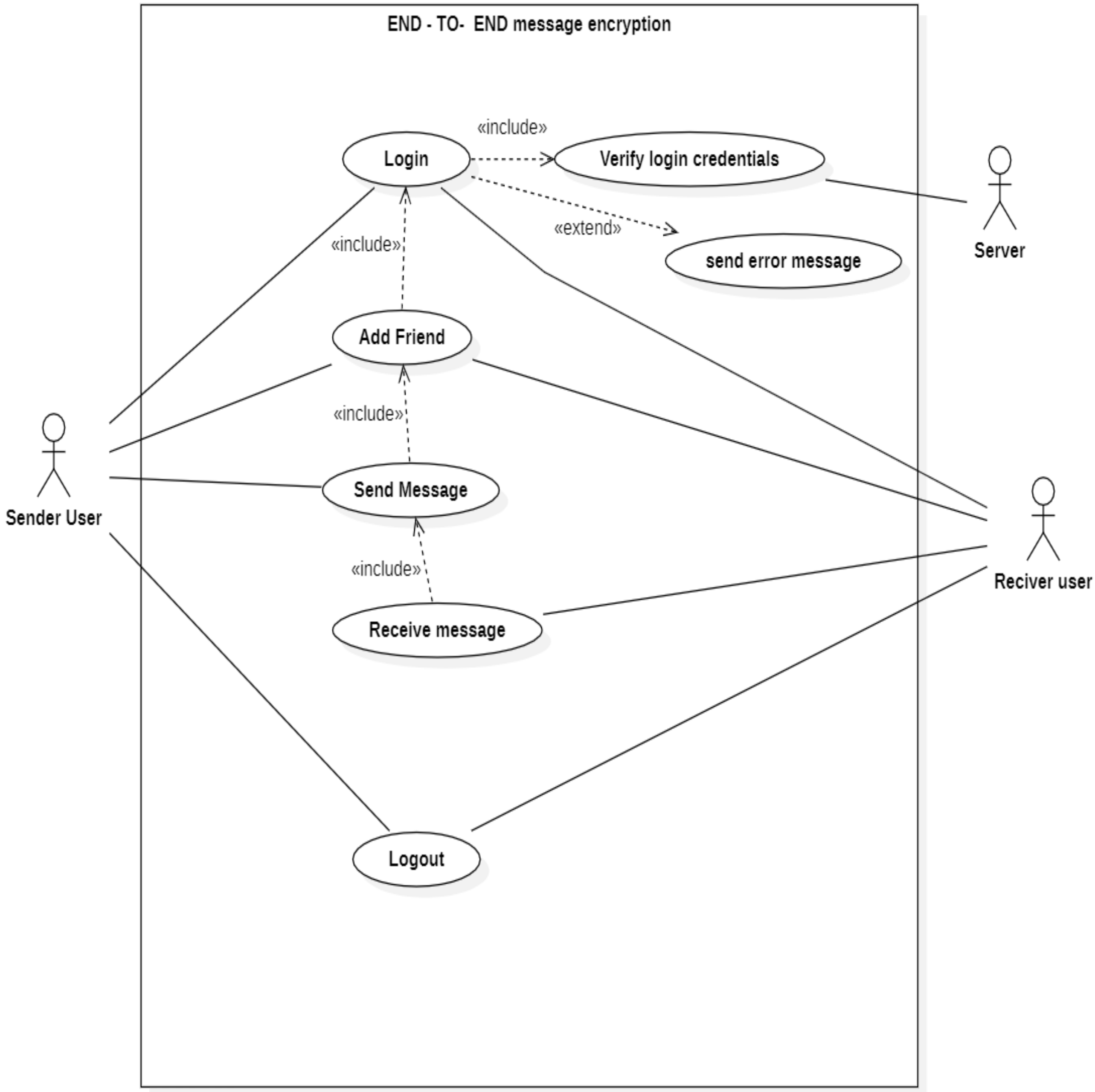
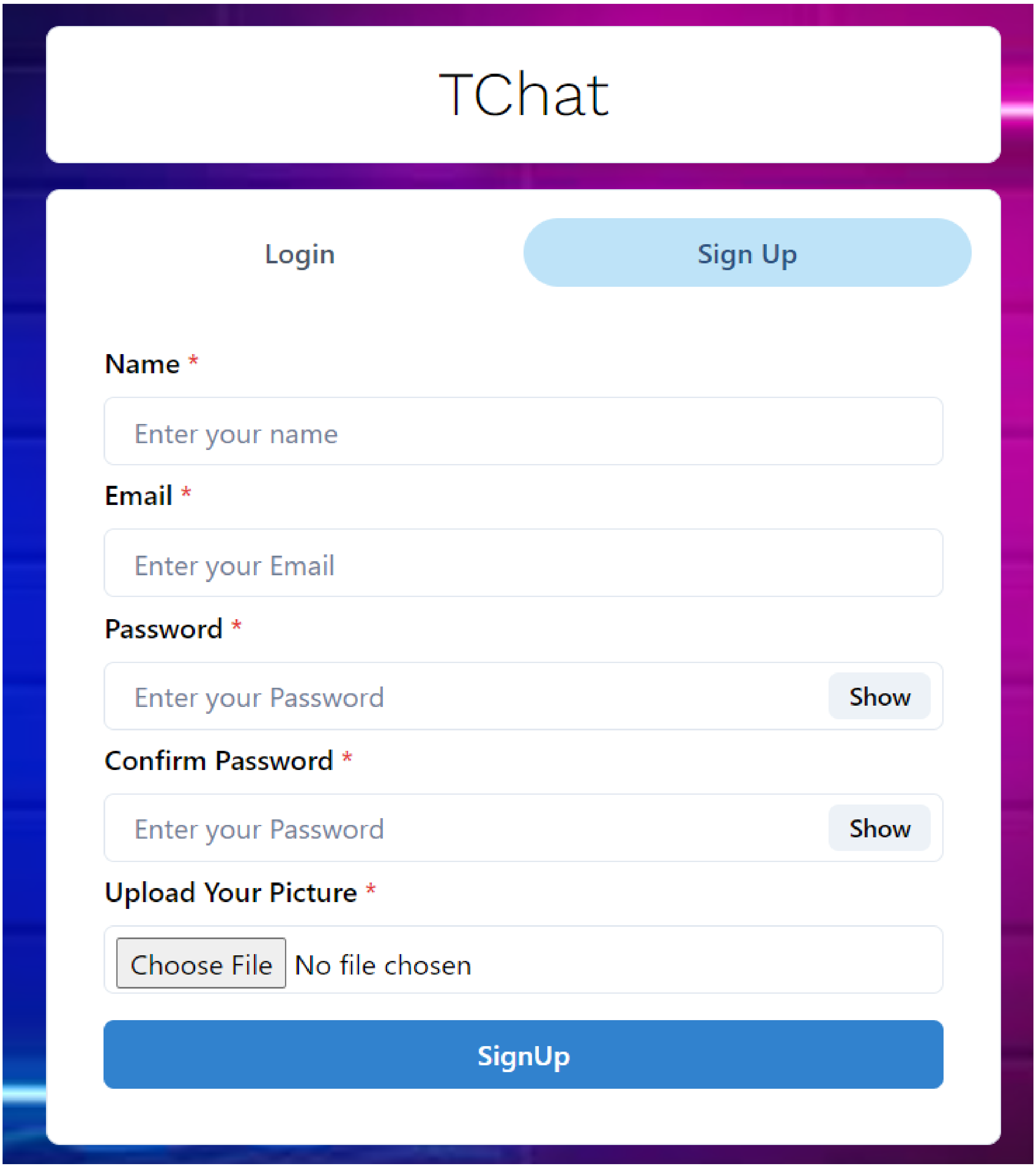


Figure 3.7: Use Case Diagram

CHAPTER 4
RESULT AND DISCUSSION

4.1 Expected Output:



TChat

Login

Sign Up

Name *

Enter your name

Email *

Enter your Email

Password *

Enter your Password Show

Confirm Password *

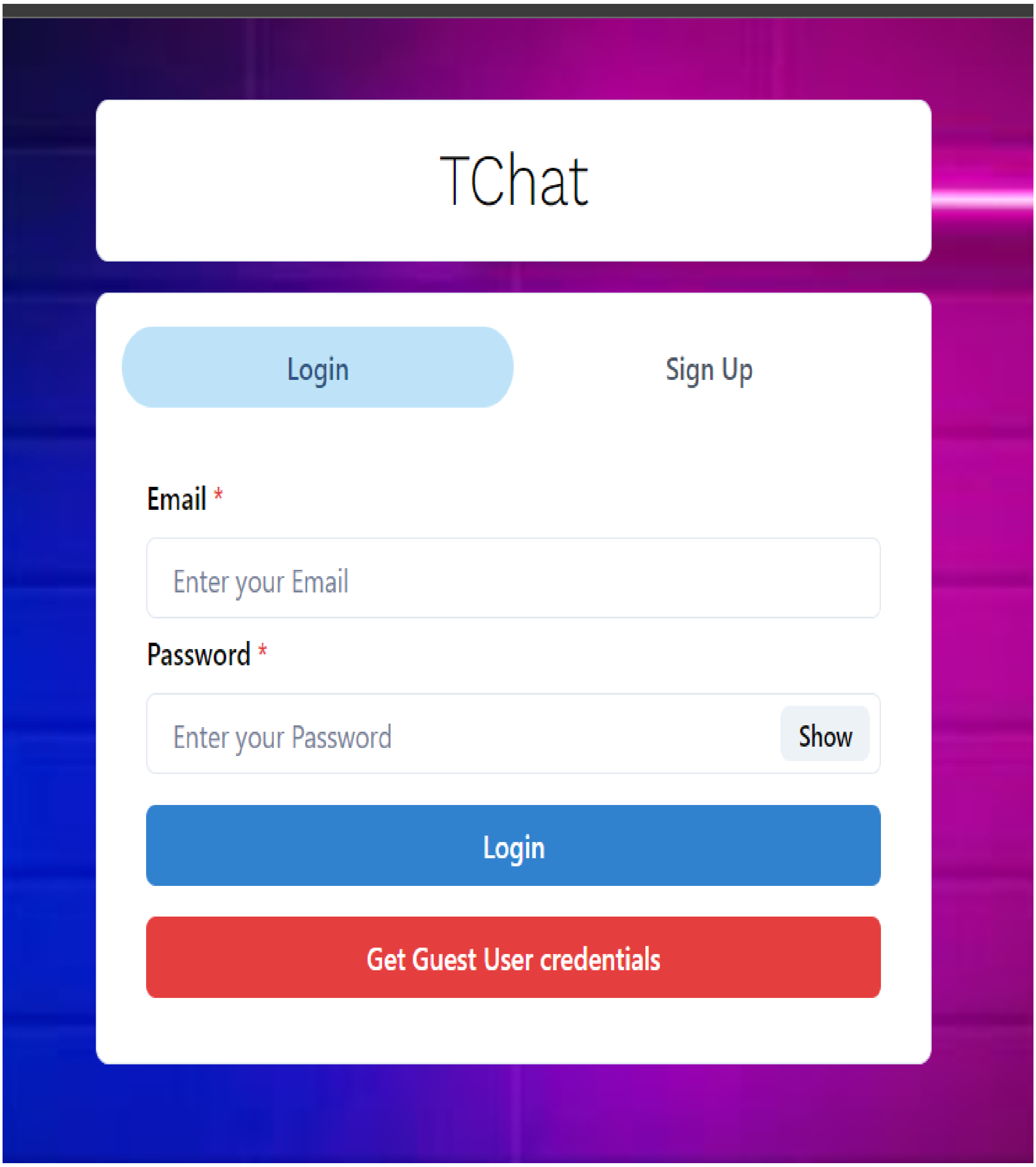
Enter your Password Show

Upload Your Picture *

Choose File No file chosen

SignUp

Figure 4.1: Sign up interface



TChat

Login

Sign Up

Email *

Enter your Email

Password *

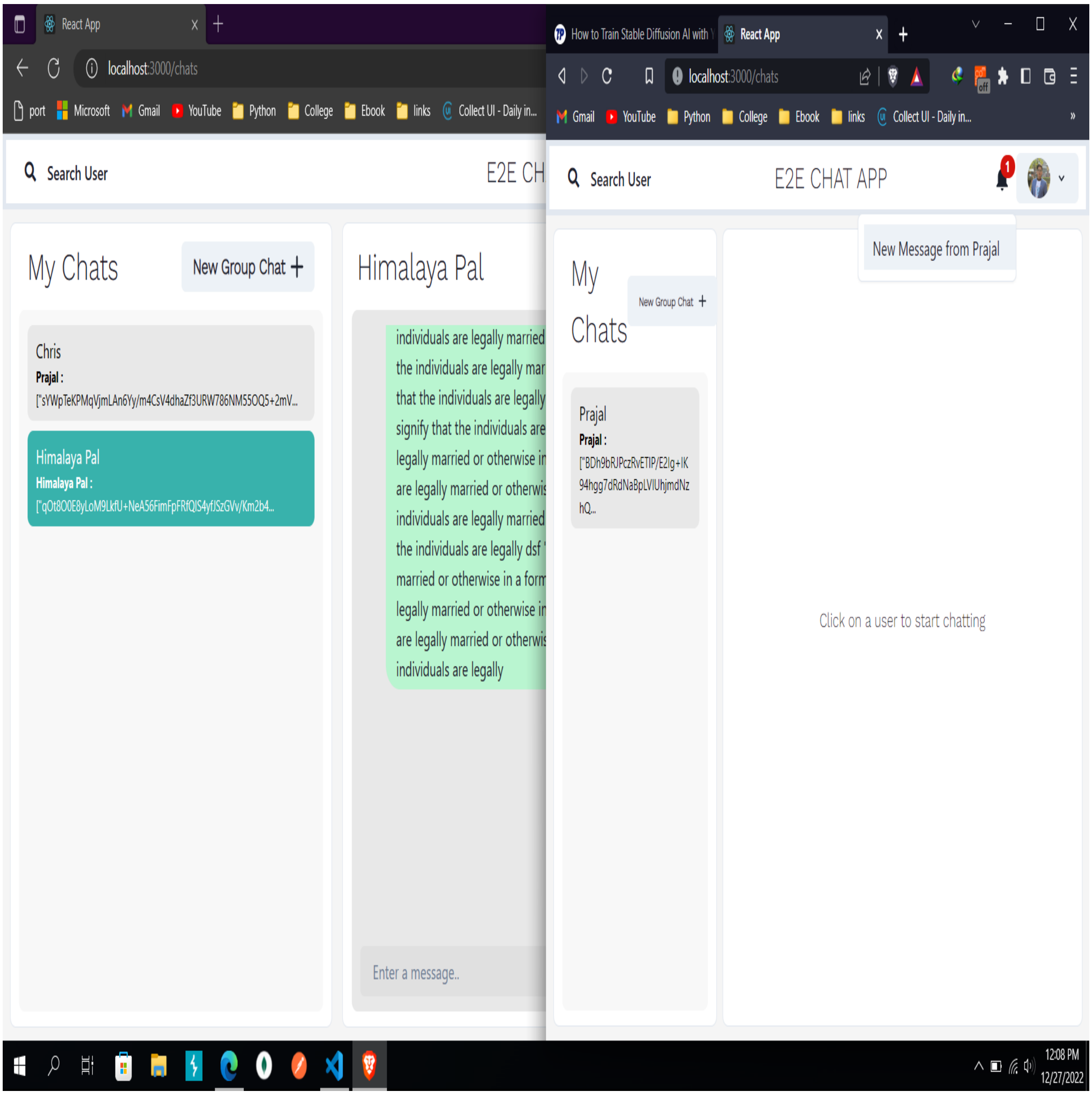
Enter your Password

Show

Login

Get Guest User credentials

Figure 4.2: Sign in interface



Chris
Prajal : ["sYWpTeKPMqVjmLAn6Yy/m4CsV4dhaZf3URW786NM55OQ5+2mV...

Himalaya Pal
Himalaya Pal : ["qOt8O0E8yLoM9LkfU+NeA56FimFpFRfQIS4yfJSzGVv/Km2b4...

individuals are legally married
the individuals are legally mar
that the individuals are legally
signify that the individuals are
legally married or otherwise in
are legally married or otherwis
individuals are legally married
the individuals are legally dsf
married or otherwise in a form
legally married or otherwise in
are legally married or otherwis
individuals are legally

Prajal
Prajal : ["BDh9bRJPczRvETIP/E2Ig+IK94hgg7dRdNaBpLVIUhjmdNzhQ...

Figure 4.3: Chat interface

4.2 Work on Progress:

The ultimate goal of this project is to develop a working model of secure end to end chat web-app. This will include a working web client for user login and chat in real time using socket.io . We have finished creating login and signup system. Our program allows to search and add contacts; create a group chat. Our program also allows Users to send enrypted messages to each other.

4.3 Work Remaining:

We are still working on the backend and frontend of the chat system. We are yet to integrate the DH algorithm to our current program. So far our program only allows encrypted chat between single users. We are yet to have encrypted group chat.

4.4 Work Schedule

Scheduling establishes the timelines, delivery and availability of project resources whether they be personnel, inventory or capital. For this reason, any project without a schedule is a project doomed to issue down the road.

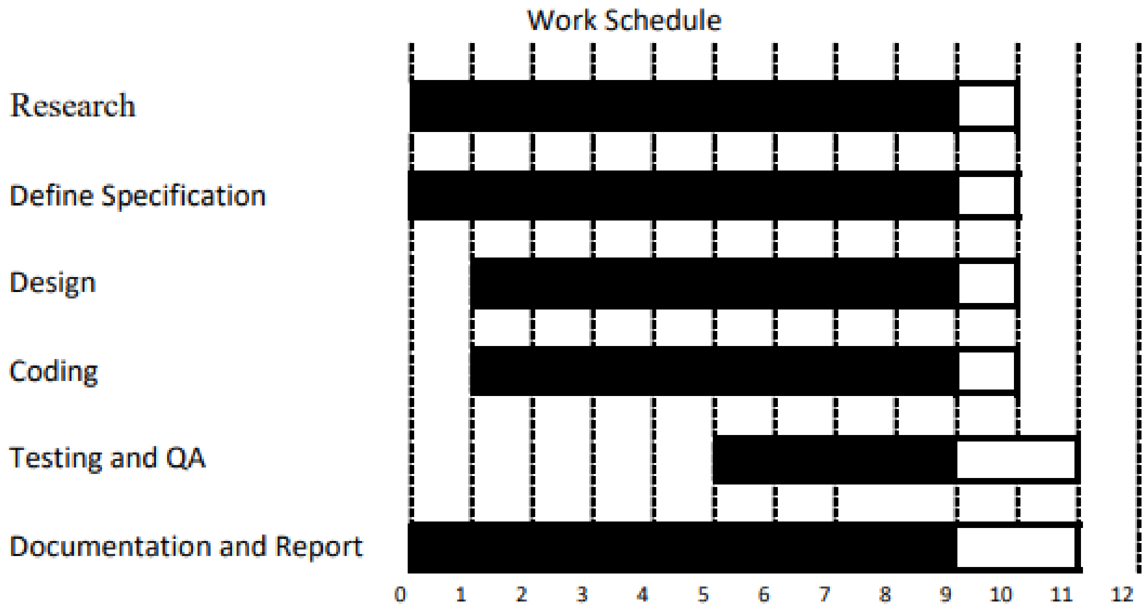


Figure 4.4: Gantt Chart

REFERENCES

[1] M. Barakat, C. Eder, and T. Hanke, “An introduction to cryptography,” 2018.

[2] F. Sönmez and M. K. Abbas, “Development of a client/server cryptography-based secure messaging system using rsa algorithm,” *J. Manag. Eng. Inf. Technol*, vol. 4, no. 6, 2017.

[3] N. Sabah, J. Mohamad, J. M. Kadhim, and B. N. Dhannoon, “Developing an end-to-end secure chat application,” 2017. [Online]. Available: https://www.researchgate.net/publication/322509087

[4] J. E. Avestro, A. M. Sison, and R. P. Medina, “Hybrid algorithm combining modified diffie hellman and rsa,” *Proceedings of the 2019 IEEE 4th International Conference on Technology, Informatics, Management, Engineering and Environment, TIME-E 2019*, pp. 100–104, 11 2019.

[5] H. Bodur and R. Kara, “Secure sms encryption using rsa encryption algorithm on android message application the development of face recognition based web authentication system view project iot and cloud based smart parking management system view project secure sms encryption using rsa encryption algorithm on android message application,” 2015. [Online]. Available: https://www.researchgate.net/publication/298298027

[6] “End-to-end encryption in chats - viber support knowledge base.” [Online]. Available: https://help.viber.com/en/article/end-to-end-encryption-in-chats

[7] “(pdf) developing an end-to-end secure chat application.” [Online]. Available: https://www.researchgate.net/publication/322509087_Developing_an_End-to-End_Secure_Chat_Application

[8] “Telegram faq.” [Online]. Available: https://telegram.org/faq#q-so-how-do-you-encrypt-data

[9] “Faq for the technically inclined.” [Online]. Available: https://core.telegram.org/techfaq#q-how-does-end-to-end-encryption-work-in-mtproto

[10] “Messenger secret conversations technical whitepaper,” 2017.