KANTIPUR ENGINEERING COLLEGE

(Affiliated to Tribhuvan University)

Dhapakhel, Lalitpur

[Subject Code: CT707]

A MAJOR PROJECT PROPOSAL ON

END-TO-END ENCRYPTION AND DECRYPTION USING CRYPTOGRAPHY (ASYMMETRIC KEY ENCRYPTION AND DECRYPTION (RSA))

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

June, 2022

END-TO-END ENCRYPTION AND DECRYPTION USING CRYPTOGRAPHY (ASYMMETRIC KEY ENCRYPTION AND DECRYPTION (RSA))

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

June, 2022

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 with file compression feature. List of requirements to make such application are provided in this paper.[1]

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 compressed the message using huffman Encoding algorithm. Then used RSA to encrypt that encoded data. The login system is also secured because we used AES library to secure the password.

In this way we can maintain the data more securely. Since we used huffman to compress the data and RSA algorithm for securing the data.[2]

Keywords – RSA, Huffman

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.[3]

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

The purpose of this project is to provide the correct data with security to the users. For some of the users the data might be lost during the transmission process in the network and for some, the data might be changed by the unauthorized person in the network and there are some other security problems in the network. Our application will give you more Security to the data present in the network and there will be able to reduce the loss of data in the network which will be transmitted from the sender to the receiver using the latest technologies. Only the Authorized persons i.e., who are using our application will be there in the Network.

The proposed algorithm is to compress the message at one end and ask the public key of another person and use that public key to encrypt the message and the person only decrypt the message using his private key.Using Huffman encoding the message compressed is loss less.

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 data 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
- It provides encryption, decryption and compression to messages.

1.6 Feasibility Analysis

1.6.1 Economic Feasibility

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

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

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. The system comprises only of farmer's end.

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
- PostgreSQL Server
- Python

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.[4]

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.

RSA algorithm, which is one of the public-key encryption methods and more reliable than the private key encryption algorithms, is used in our proposed app for secure messaging.[5]

RSA algorithm is asymmetric cryptography algorithm. Asymmetric means it works in two different keys: Public and Private. Public key is given to everyone and Private key is kept private.

The idea of RSA is based on the fact that it is difficult to factorize a large integer. The public key consists of two numbers where one number is multiplication of two large prime numbers. And private key is also derived from the same two prime numbers. So if somebody can factorize the large number, the private key is compromised. Therefore encryption strength totally lies on the key size and if we double or triple the key size, the strength of encryption increases exponentially. RSA keys can be typically 1024 or 2048 bits long, but experts believe that 1024 bit keys could be broken in the near future.

But till now it seems to be an infeasible task.[6]

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 .[7] 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.

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 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 [8]

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 .

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 Hellman Algorithm Steps:

- The first party picks two prime numbers, g and p and tells them to the second party.
- The second party then picks a secret number (let's call it a), and then it computes g^a mod p and sends the result back to the first party; let's call the result A. Keep in mind that the secret number is not sent to anyone, only the result is.
- Then the first party does the same; it selects a secret number b and calculates the result B=g^b mod p similar to the step 2.
- Then, this result is sent to the second party.
- The second party takes the received number B and calculates K_a= B^a mod p
- The first party takes the received number A and calculates K_b = A^b mod p.
- The

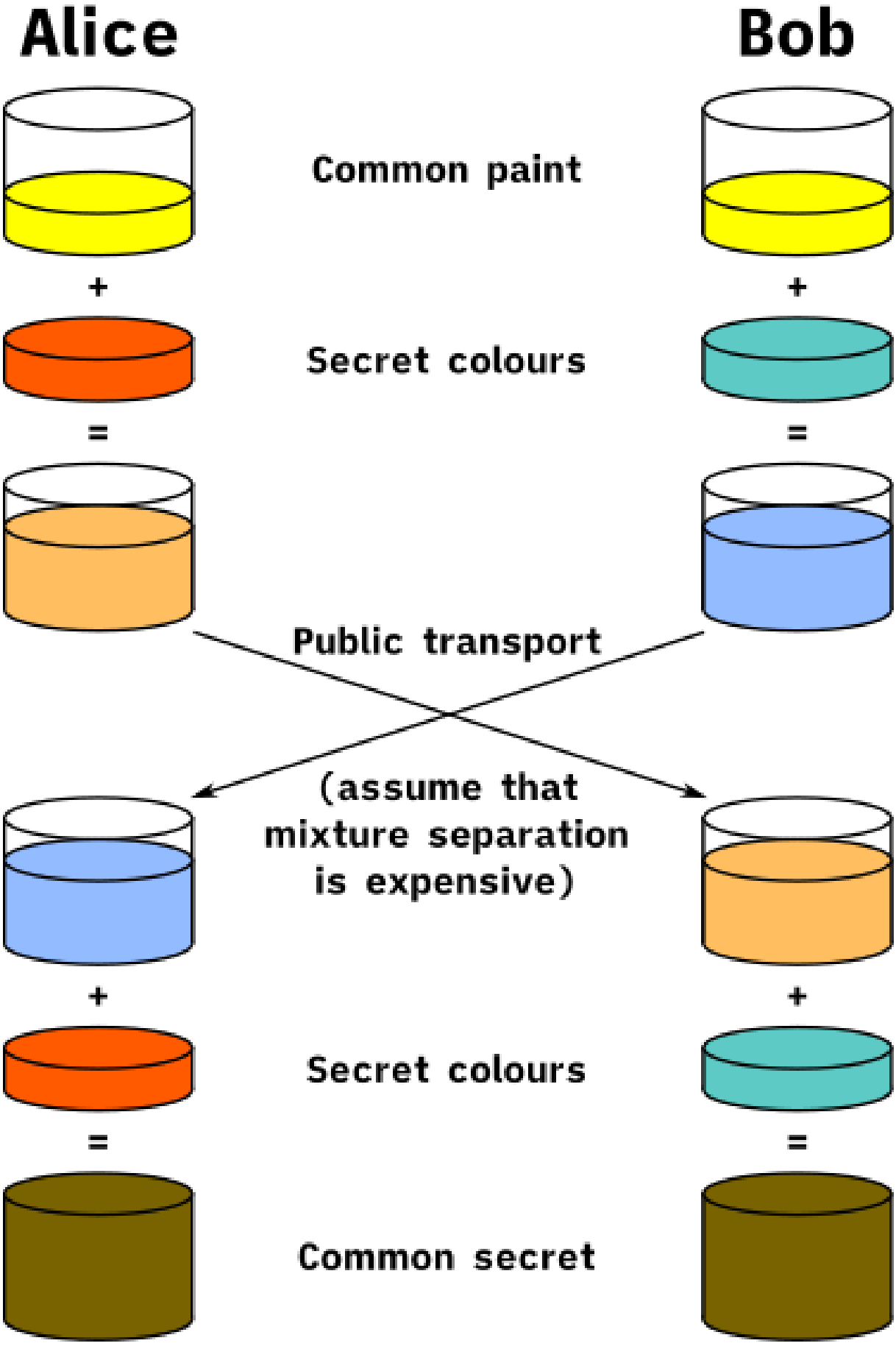

Figure 3.1: Diffie Hellman working mechanism

3.1.3 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,

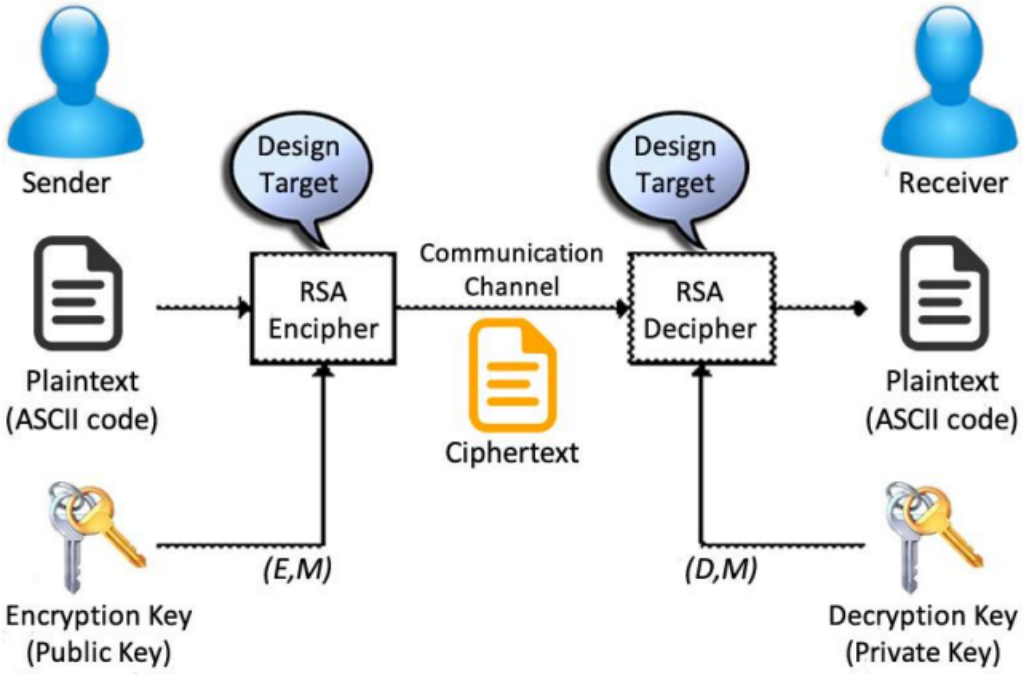

Figure 3.2: RSA algorithm working mechanism

3.1.4 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 $M = C^d \bmod n$
- A text encrypted with the public key can only be solved with the private key

3.1.5 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:

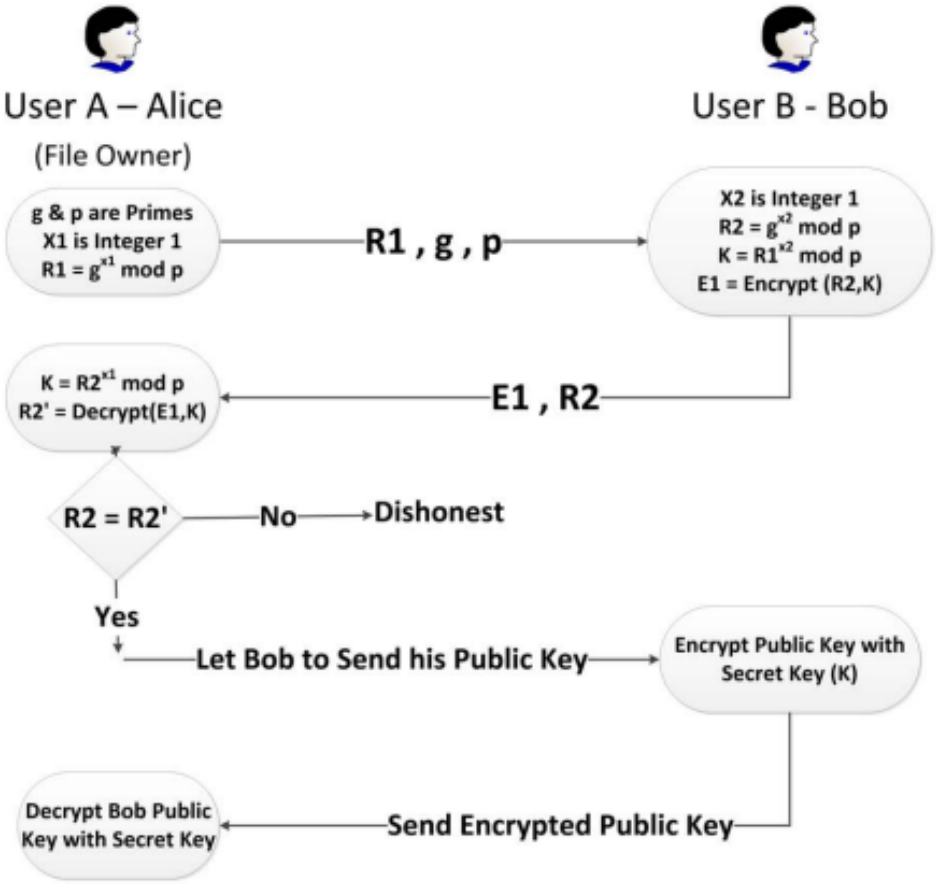

Figure 3.3: User authentication and encryption of public key

3.1.6 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 $K_{Bob} = R_1^{x_2} mod p$ and $E_1 = Encrypt(R_2, K_{Bob})$ and sends R_2, E_1 to Alice. After that, Alice computes $K_{Alice} = R_2^{x_1} mod p$ 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.

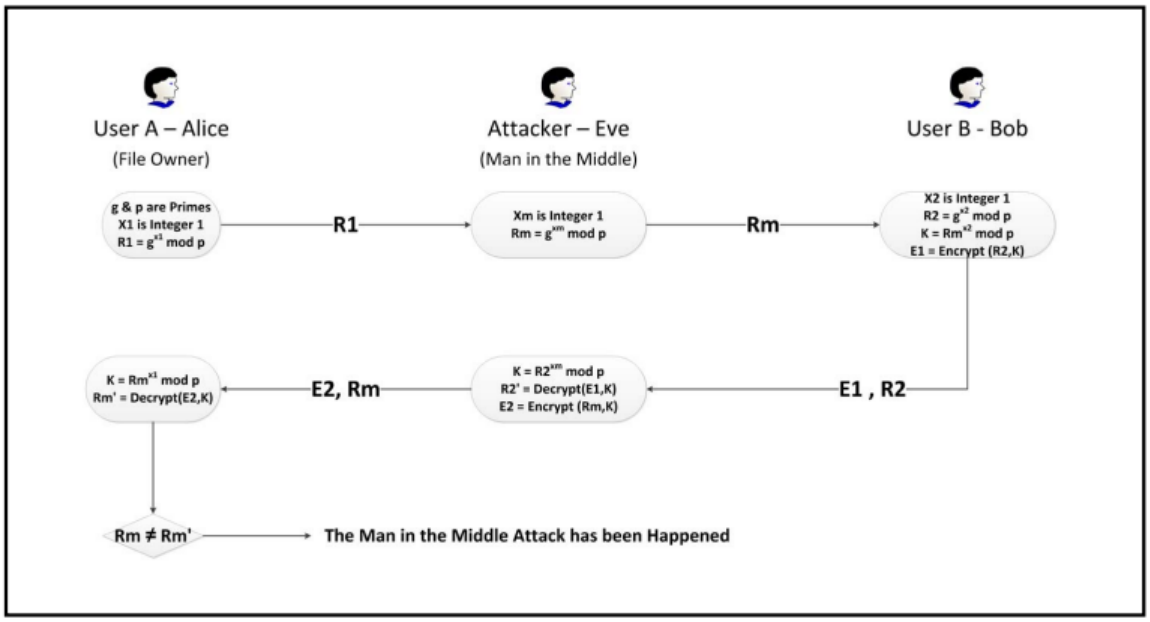

Figure 3.4: Man in the Middle Attack Prevention by Proposed Algorithm

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.

3.1.7 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 use Encryption to protect the data.

The next thing to talk about is integrity. Well, the idea here is to make sure that data has not been modified. Corruption of data is a failure to maintain data integrity. We verified that we are sending message to is a genuine person so no tempering 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

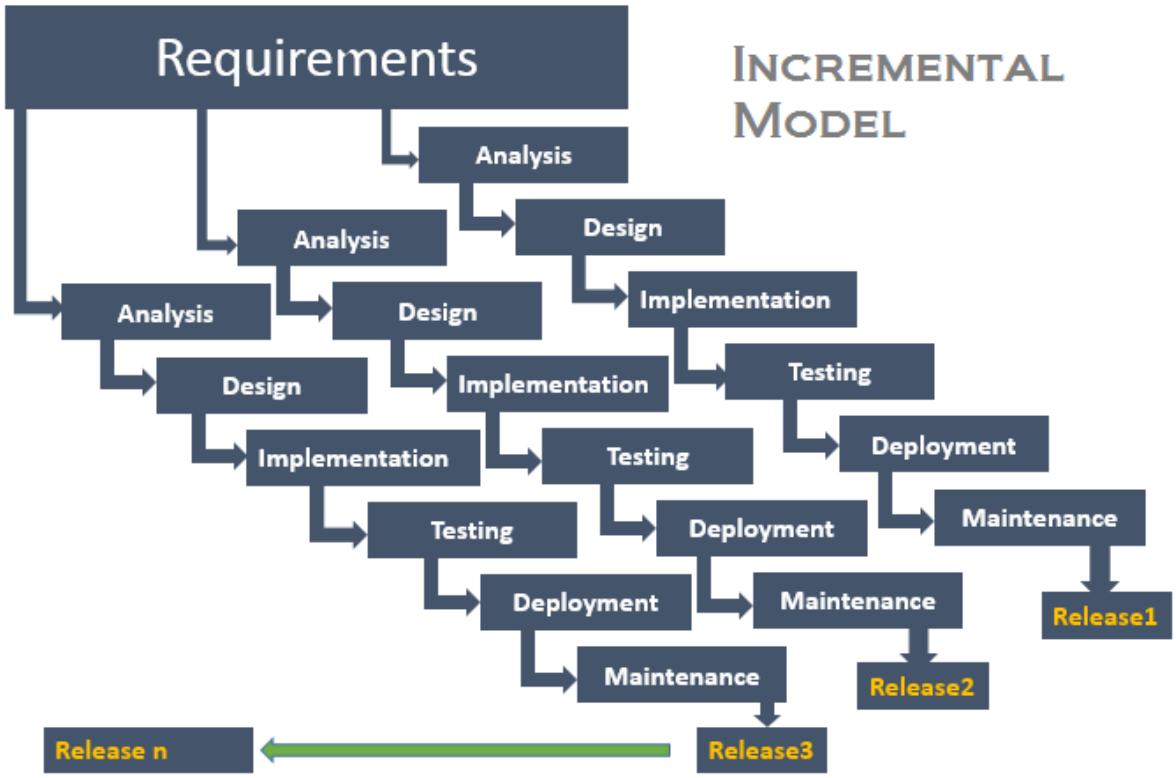

Figure 3.5: Incremental Model Block Diagram

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:

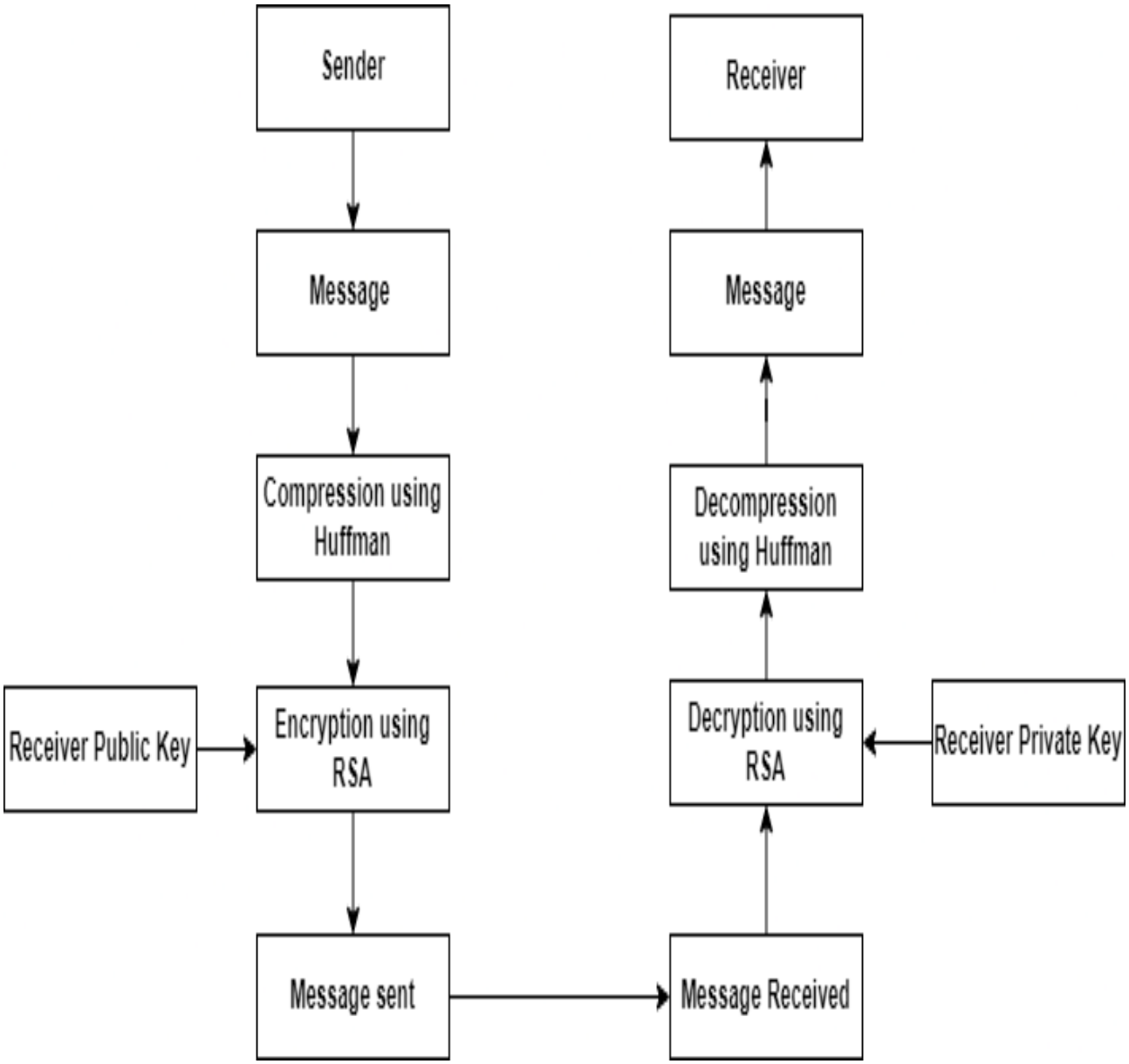

Figure 3.6: Block Diagram

3.4 Use Case Diagram:

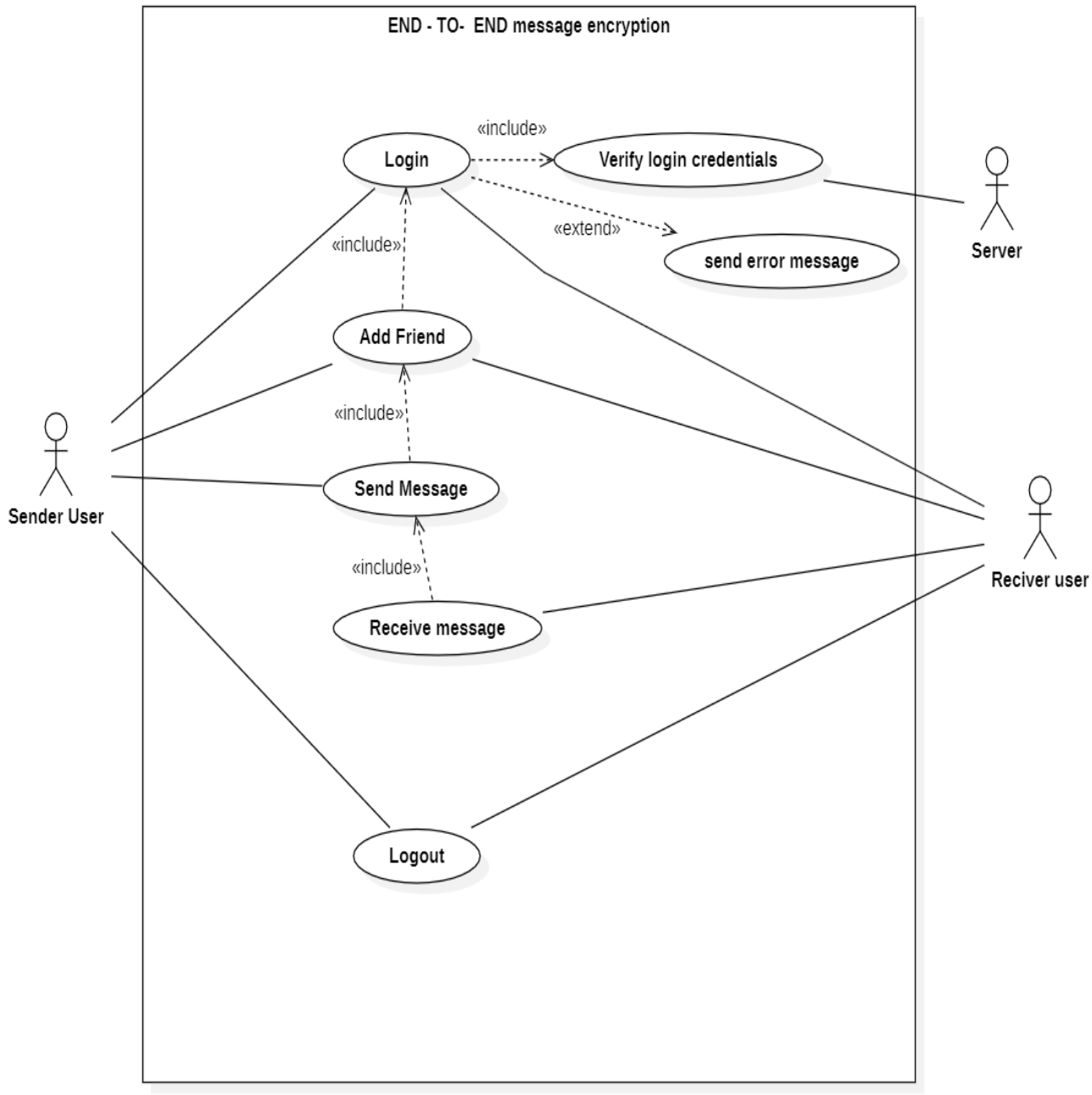

Figure 3.7: Use Case Diagram

CHAPTER 4
EPILOGUE

4.1 Expected Output:

The application will have a good user interface and user experience. It will be a web based application. In this application the user will first need to provide their details in order to login. Then the user can send messages to receiver which is first compressed using huffman and the gets encrypted using RSA algorithm. After that same message is sent to receiver which gets decrypted using RSA algorithm and the decompressed using huffman and receiver receives message with higher security and privacy. In this the message is encrypted with public key of receiver and the decrypted using private key of receiver. We expect a clean UI of this message communication web application with higher security and privacy.

4.2 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.

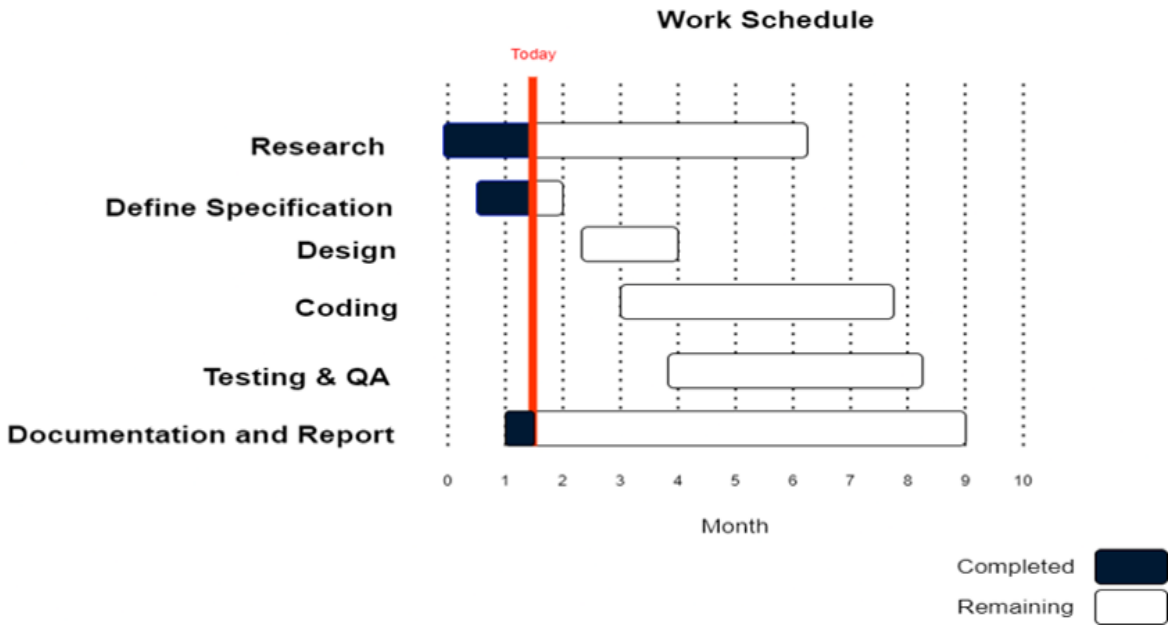

Figure 4.1: Gantt Chart

REFERENCES

[1] P. Chauhan, S. Gupta, and N. S. Mathur, "A new approach to encryption using huffman coding," pp. 76–82, 2016. [Online]. Available: http://ijpsat.ijsht-journals.org

[2] J. K. Dawson, F. Twum, J. B. H. Acquah, and B. K. Ayawli, "An enhanced rsa algorithm for data security using gaussian interpolation formula," 2022. [Online]. Available: https://doi.org/10.21203/rs.3.rs-1326669/v1

[3] M. Barakat, C. Eder, and T. Hanke, "An introduction to cryptography," 2018.

[4] N. Sabah, J. Mohamad, and B. N. Dhannoon, "Developing an end-to-end secure chat application," vol. 17, 11 2017.

[5] H. Bodur and R. Kara, "Secure sms encryption using rsa encryption algorithm on android message application," 06 2015.

[6] "Rsa algorithm in cryptography - geeksforgeeks." [Online]. Available: https://www.geeksforgeeks.org/rsa-algorithm-cryptography/

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

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