

SECURED DATA TRANSFER WITH MULTI LAYERED SECURED ENCRYPTION STANDARDS USING ONION PROTOCOL

A PROJECT REPORT

Submitted by

MOHAN RAJ S (211419205106)

JAGADESH E (211419205072)

GANESH N (211419205050)

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

INFORMATION TECHNOLOGY

PANIMALAR ENGINEERING COLLEGE, POONAMALLEE

ANNA UNIVERSITY : CHENNAI 600 025

APRIL 2023

ANNA UNIVERSITY: CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report **“SECURED DATA TRANSFER WITH MULTI LAYERED SECURED ENCRYPTION STANDARDS USING ONION PROTOCOL”** is the bonafide work of **“MOHAN RAJ S (211419205106), JAGADESH E (211419205072), GANESH N (211419205050)”** who carried out the project under my supervision.

SIGNATURE

Dr. M. HELDA MERCY M.E., Ph.D.,

HEAD OF THE DEPARTMENT

Department of Information Technology

Panimalar Engineering College

Poonamallee, Chennai - 600 123

SIGNATURE

Dr. M. SUMITHRA M.E., Ph.D.,

SUPERVISOR

ASSOCIATE PROFESSOR

Department of Information Technology

Panimalar Engineering College

Poonamallee, Chennai - 600 123

Submitted for the project and viva-voce examination held on _____

SIGNATURE

INTERNAL EXAMINER

SIGNATURE

EXTERNAL EXAMINER

DECLARATION

I hereby declare that the project report entitled “**SECURED DATA TRANSFER WITH MULTI LAYERED SECURED ENCRYPTION STANDARDS USING ONION PROTOCOL**” which is being submitted in partial fulfilment of the requirement of the course leading to the award of the ‘Bachelor Of Technology in Information Technology ’ in **Panimalar Engineering College, Autonomous institution Affiliated to Anna university- Chennai** is the result of the project carried out by me under the guidance of **Dr. M. SUMITHRA M.E., Ph.D., Associate Professor in the Department of Information Technology**. I further declared that I or any other person has not previously submitted this project report to any other institution/university for any other degree/ diploma or any other person.

Date:

MOHAN RAJ S

Place: Chennai

JAGADESH E

GANESH N

It is certified that this project has been prepared and submitted under my guidance.

Date:

Dr. M. SUMITHRA M.E., Ph.D.,

Place: Chennai

(Associate Professor / IT)

ACKNOWLEDGEMENT

A project of this magnitude and nature requires kind co-operation and support from many, for successful completion . We wish to express our sincere thanks to all those who were involved in the completion of this project.

Our sincere thanks to **Our Honorable Secretary and Correspondent, Dr. P. CHINNADURAI, M.A., Ph.D.,** for his sincere endeavor in educating us in his premier institution.

We would like to express our deep gratitude to **Our Dynamic Directors, Mrs. C. VIJAYA RAJESHWARI and Dr. C. SAKTHI KUMAR, M.E., M.B.A., Ph.D., and Dr. SARANYA SREE SAKTHIKUMAR .,B.E., M.B.A., Ph.D.,** for providing us with the necessary facilities for completion of this project.

We also express our appreciation and gratefulness to **Our Principal Dr. K. MANI, M.E., Ph.D.,** who helped us in the completion of the project. We wish to convey our thanks and gratitude to our head of the department, **Dr. M. HELDA MERCY, M.E., Ph.D.,** Department of Information Technology, for her support and by providing us ample time to complete our project.

We express our indebtedness and gratitude to our Project co-ordinator **Mr. M. DILLI BABU, M.E.,(Ph.D.,)** Associate Professor, Department of Information Technology for his guidance throughout the course of our project. We also express sincere thanks to our supervisor **Dr. M. SUMITHRA M.E., Ph.D.,** Associate Professor, Department of Information Technology for providing the support to carry out the project successfully. Last, we thank our parents and friends for providing their extensive moral support and encouragement during the course of the project.

ABSTRACT

Because of the discoveries of global-scale widespread monitoring operations, Internet users are increasingly concerned about their privacy. Nevertheless, for network service providers, this is normally undesirable since attackers would be able to anonymize themselves and bypass regulation while executing network attacks. As a result, A multi-layered secured encryption standard using onion protocol was proposed in this project. In order to ensure secure data transfer within the, we develop a robust security mechanism that also collects information about the packets that surrounding nodes are transmitting. After the best route has been chosen after considering each node, just the packets are sent. Each node that joins the network is given two keys. We also use a multi-level encryption framework in this project. Our project would first choose the optimum path to every site where data has to be delivered. The data is first encrypted at the source end using the RSA technique, and once a path has been found, it is then sent to the subsequent node, which finally goes to the destination. Assuming A and B are the nodes indentified as intermediate nodes to reach the destination from Source node. So now the Path will S, A, B, D (S – Source, D – Destination). Source node first encrypt the Data using RSA Algorithm, then same the data is encrypted using D's Key 1, followed by encryption using B's K1. Now the same Encrypted data is again encrypted using A's K1. Now this wholesome encrypted data is sent to "A" Node. A node will provide its K2 and that part layer is decrypted. Now this Encrypted data is sent to B. Now B Node will provide its K2 and that part layer is decrypted. Now the finally the encrypted data is sent to the destination. In the destination, D will provide its K2 to decrypt. Now the single Encrypted data will be left out. The initial Decrypted Key is transferred to the destination from the source, which is used to decrypt the final data in the destination.

TABLE OF CONTENTS

| CHAPTER NO | TITLE | PAGE NO |
|---------------|--|------------|
| | ABSTRACT | v |
| | LIST OF TABLES | ix |
| | LIST OF FIGURES | x |
| | LIST OF ABBREVIATIONS | xii |
| 1 | INTRODUCTION | 1 |
| | 1.1 OVERVIEW OF THE PROJECT | 1 |
| | 1.2 NEED FOR THE PROJECT | 3 |
| | 1.3 OBJECTIVE OF THE PROJECT | 4 |
| | 1.4 SCOPE OF THE PROJECT | 5 |
| 2 | LITERATURE SURVEY | 7 |
| 3 | SYSTEM ANALYSIS | 17 |
| | 3.1 EXISTING SYSTEM | 17 |
| | 3.1.1 Disadvantages Of Existing System | 17 |
| | 3.2 PROPOSED SYSTEM | 17 |
| | 3.2.1 Advantages Of Proposed System | 18 |
| 4 | REQUIREMENT SPECIFICATION | 19 |
| | 4.1 HARDWARE REQUIREMENTS | 19 |
| | 4.2 SOFTWARE REQUIREMENTS | 19 |
| | 4.3 REQUIREMENT ANALYSIS | 19 |
| | 4.3.1 Functional Requirements | 19 |
| | 4.3.2 Non-Functional Requirements | 21 |

| | | |
|----------|-----------------------------------|-----------|
| 5 | SYSTEM DEVELOPMENT | 22 |
| | 5.1 SYSTEM ARCHITECTURE | 22 |
| | 5.2 UML DIAGRAM | 24 |
| | 5.2.1 Use Case Diagram | 25 |
| | 5.2.2 Class Diagram | 26 |
| | 5.2.3 Sequence Diagram | 27 |
| | 5.2.4 Collaboration Diagram | 28 |
| | 5.2.5 Activity Diagram | 29 |
| | 5.2.6 Entity Relationship Diagram | 30 |
| | 5.2.7 Data Flow Diagram | 31 |
| | 5.3 FEASIBILITY STUDY | 32 |
| 6 | MODULES | 33 |
| | 6.1 NETWORK CONSTRUCTION | 33 |
| | 6.2 FIRST LEVEL ENCRYPTION | 35 |
| | 6.3 MULTI LAYER ENCRYPTION MODEL | 37 |
| | 6.4 DECRYPTION PROCESS | 40 |
| 7 | IMPLEMENTATION | 42 |
| | 7.1 SAMPLE CODE | 42 |
| | 7.2 SAMPLE SCREEN SHOTS | 56 |
| 8 | TESTING AND MAINTANENCE | 64 |
| | 8.1 BLACK BOX TESTING | 65 |
| | 8.2 WHITE BOX TESTING | 65 |
| | 8.3 UNIT TESTING | 66 |
| | 8.4 INTEGRATION TESTING | 66 |
| | 8.5 SYSTEM TESTING | 66 |
| | 8.6 ACCEPTANCE TESTING | 67 |

| | | |
|----------|---|----|
| | 8.7 TEST CASE SPECIFICATION | 68 |
| 9 | CONCLUSION AND FUTURE ENHANCEMENTS | 69 |
| | 9.1 CONCLUSION | 69 |
| | 9.2 FUTURE ENHANCEMENTS | 69 |
| | REFERENCES | 70 |

LIST OF TABLES

| TABLE NO | NAME OF THE TABLE | PAGE NO |
|---------------------|--|--------------------|
| 1.1 | Comparison of Onion Protocol With Other Algorithms | 2 |
| 8.1 | Test Case Specification | 68 |

LIST OF FIGURES

| FIGURE NO | NAME OF THE FIGURE | PAGE NO |
|------------------|---------------------------------|----------------|
| 3.1 | Flow Diagram Of Proposed System | 18 |
| 5.1 | System Architecture | 22 |
| 5.2 | Use Case Diagram | 25 |
| 5.3 | Class Diagram | 26 |
| 5.4 | Sequence Diagram | 27 |
| 5.5 | Collaboration Diagram | 28 |
| 5.6 | Activity Diagram | 29 |
| 5.7 | Entity Relationship Diagram | 30 |
| 5.8 | Level 0 DFD Diagram | 31 |
| 5.9 | Level 1 DFD Diagram | 31 |
| 6.1 | Network Construction | 34 |
| 6.2 | First Layer Encryption | 36 |
| 6.3 | RSA Algorithm | 38 |
| 6.4 | SHA-256 Algorithm | 39 |
| 6.5 | Data Decryption | 41 |
| 7.1 | Node Login | 56 |
| 7.2 | New Node Creation | 56 |
| 7.3 | Key Generation | 57 |

| | | |
|------|---|----|
| 7.4 | Node Login Success | 57 |
| 7.5 | Node Configuration | 58 |
| 7.6 | Node Connection | 58 |
| 7.7 | File Selection | 59 |
| 7.8 | Encryption by Source Node | 59 |
| 7.9 | File After Encryption | 59 |
| 7.10 | Selection of Destination Node | 60 |
| 7.11 | Best Path Selection | 60 |
| 7.12 | Decryption by Intermediate Node | 61 |
| 7.13 | Intermediate Node ID Verification | 61 |
| 7.14 | Encrypted Data Received at Destination Node | 62 |
| 7.15 | Destination Decryption Key Verification | 62 |
| 7.16 | Decryption Key Verification | 63 |
| 7.17 | Original Data at Destination Node | 63 |

LIST OF ABBREVIATIONS

| | |
|-----|------------------------------------|
| RSA | Rivest, Shamir, Adleman |
| SHA | Secure Hash Algorithm |
| AES | Advanced Encryption Standard |
| DES | Data Encryption Standard |
| ECC | Elliptical curve cryptography |
| VPN | Virtual private network |
| IP | Internet Protocol |
| DNS | Domain Name System |
| RAM | Random Access Memory |
| HDD | Hard Disk Drive |
| IDE | Integrated Development Environment |
| JDK | Java Development Kit |
| UML | Unified Modeling Language |
| ER | Entity Relationship |
| DFD | Data Flow Diagram |

INTRODUCTION

1.1 OVERVIEW OF THE PROJECT

In any multi hop wireless network communication, when a node sends a packet to the destination, it relies on the intermediate nodes to relay the packets. This multi hop packet transmission can extend the network coverage area using limited power. The proper data transfer is depends on the multiple parameters such as Energy Levels of the intermediate Nodes and also based on its Secured Processing. Only if the Intermediate Hops / Nodes are secured and cannot be hacked, all the packets can be transferred safely to the Destination from the Source Node. We also consider heterogeneous multi hop wireless networks, where the nodes' mobility level and hardware / Energy resources may vary greatly. In this project, we propose a best Process for the secured data transfer between Source and Destination node safely. We collect every Intermediate Node's ID for verification. Every Node will 2 Keys during Registration. Data is first encrypted using RSA Algorithm and then by the Intermediate Node's Key 1. Before sending the Packets we need to verify the Capacity of that Node also. Multi Layer Encryption Technique is applied for the highly secured data transfer. Every Node's ID and Key 2 is verified only then the encrypted data is send to that node for transfer.

Multi-layer encryption, also known as nested encryption, is a security technique that involves using multiple layers of encryption to protect data. Each layer of encryption adds an additional level of security, making it more difficult for unauthorized users to access or decrypt the data. In a multi-layer encryption scheme, the data is encrypted with one algorithm, and then the encrypted data is encrypted again with another algorithm. The result is an encrypted data file that is virtually impossible to decipher without the proper decryption keys. One advantage of multi-layer encryption is that it provides an extra layer of security in case one layer is compromised. For example, if an attacker manages to break

through the first layer of encryption, they will still have to deal with the second layer of encryption, which adds an extra level of protection.

Table 1.1 Comparison of onion protocol with other algorithms

| Routing Scheme | Efficiency | Security | Speed |
|-----------------------|-------------------|-----------------|--------------|
| Onion Routing | High | High | Moderate |
| Tor | High | High | Moderate |
| VPN | High | Moderate | High |
| IP Routing | High | Low | High |
| DNS Routing | Low | Low | High |

Another advantage of multi-layer encryption is that it allows organizations to comply with multiple data protection regulations. For instance, if a company operates in a jurisdiction with strict data protection laws, it may need to encrypt data using a specific algorithm. By using a multi-layer encryption scheme, the company can comply with those regulations while still using other encryption algorithms to provide additional layers of protection. However, multi-layer encryption can also introduce complexity and slow down data processing times. It can also increase the risk of key management issues, as

multiple keys may be needed to decrypt the data. Therefore, it is essential to carefully consider the trade-offs and risks before implementing a multi-layer encryption scheme.

1.2 NEED FOR THE PROJECT

In today's world, data is one of the most valuable assets. Companies, organizations, and individuals all store sensitive information that needs to be protected from unauthorized access. One of the most common methods of protecting data is encryption. Encryption is the process of converting plaintext (unencrypted data) into cipher text (encrypted data) using an algorithm and a key. The cipher text can only be decrypted back into plaintext using the same algorithm and key.

While encryption is an effective method of protecting data, it is not fool proof. Cybercriminals are constantly developing new ways to crack encryption algorithms and steal sensitive information. This is where multilayer encryption comes in. Multilayer encryption is a method of applying multiple encryption algorithms to the same data. This makes it much more difficult for hackers to crack the encryption and steal the data.

There are several reasons why multilayer encryption is necessary for certain projects. First, some projects involve highly sensitive information that needs to be protected at all costs. For example, government agencies may store classified information that could be used to compromise national security. In this case, a single encryption algorithm may not be enough to protect the data from sophisticated hackers.

Some industries are highly regulated and require strict data protection measures. For example, the healthcare industry is subject to the Health Insurance Portability and Accountability Act (HIPAA), which requires all patient information to be protected by strong encryption. In this case, multilayer encryption may be necessary to ensure compliance with these regulations.

Some projects involve data that is valuable to multiple parties. For example, a financial institution may store data on behalf of its clients. This data may be valuable to both the financial institution and the clients. In this case, multilayer encryption can provide an additional layer of protection to ensure that the data is not compromised.

Some projects involve data that is valuable for a long period of time. For example, historical records may contain sensitive information that needs to be protected for decades or even centuries. In this case, multilayer encryption can help ensure that the data remains secure for the entire duration of its lifetime.

Some projects involve data that is transmitted over insecure networks. For example, a company may need to transmit sensitive information over the internet. In this case, multilayer encryption can provide an additional layer of protection to ensure that the data is not intercepted by unauthorized parties.

1.3 OBJECTIVE OF THE PROJECT

In this project we implement secured data transfer in the Network between source and the destination. We ensure data is transferred securely among multiple nodes in between to reach the destination. We also detect the Node's ID and its Primary key only then the Encrypted Packets will reach that node for Transfer. Multilayer encryption is a method of securing data by using multiple layers of encryption. Each layer adds an additional level of security, making it difficult for attackers to intercept or eavesdrop on the data being transmitted. This method of encryption is used in a variety of applications, including email, messaging apps, and file sharing services.

Multilayer encryption provides a high level of security for data being transmitted over the internet. The use of multiple layers of encryption makes it difficult for attackers to intercept or eavesdrop on the data, and provides protection against a variety of attacks, including man-in-the-middle attacks and replay attacks. In conclusion, multilayer encryption is a method of securing data

by using multiple layers of encryption. Each layer adds an additional level of security, making it difficult for attackers to intercept or eavesdrop on the data being transmitted. The use of multilayer encryption is becoming increasingly important as the amount of data being transmitted over the internet continues to grow. By using multiple layers of encryption, data can be protected from a variety of attacks, ensuring that it remains confidential and secure.

1.4 SCOPE OF THE PROJECT

The scope of the onion protocol for multilayer encryption is broad and encompasses a wide range of applications. One of the primary areas of application is in the realm of online privacy and security. The onion protocol is frequently used to protect online communications such as email, messaging apps, and file sharing services. This is particularly important in the era of mass surveillance, where governments and other organizations may attempt to intercept or monitor online communications. Another area of application for the onion protocol is in the realm of online anonymity.

The onion protocol is used to power the Tor network, which allows users to browse the internet anonymously. The Tor network works by routing internet traffic through a series of encrypted relays, making it difficult for anyone to trace the origin of the traffic. This has become particularly important for individuals living in countries with repressive governments, where online anonymity can be a matter of life or death. The onion protocol is also used in the realm of online commerce. The protocol is frequently used to protect online transactions, such as those that occur on e-commerce websites. By using multilayer encryption, e-commerce websites can ensure that sensitive customer data such as credit card numbers and addresses are kept secure during the transaction process.

The use of the onion protocol is not limited to online applications, however. The protocol is also frequently used in the realm of computer and network security. By using multilayer encryption, network administrators can ensure that their networks are secure from a variety of attacks, including those that seek to intercept or eavesdrop on network traffic.

Overall, the scope of the onion protocol for multilayer encryption is vast and encompasses a wide range of applications. From online privacy and security to computer and network security, the onion protocol is a powerful tool for protecting sensitive information and ensuring that online communications are kept confidential and secure. As the amount of data being transmitted over the internet continues to grow, the use of multilayer encryption will only become more important, making the onion protocol a critical tool for ensuring online privacy and security.

LITERATURE SURVEY

2.1 MULTI-LAYER ENCRYPTION TECHNIQUES FOR SECURE COMMUNICATION SYSTEMS

Author: H. Singh and M. Singh.

Year: 2016

Description

It propose a multilayer encryption approach that combines these techniques to provide a higher level of security. This can be helpful for readers who want to understand the different options available and how they can be combined to create a more secure communication system. Moreover, the paper provides a detailed comparison of the proposed multilayer encryption scheme with other existing encryption schemes and shows that the proposed approach is more secure and efficient.

Advantages

The paper presents a comprehensive review of various multilayer encryption techniques that can be used to enhance the security of communication systems. It proposes a multilayer encryption approach that combines these techniques to provide a higher level of security.

Disadvantages

One potential disadvantage of this paper is that it is mainly focused on theoretical aspects of multilayer encryption and does not provide a practical implementation of the proposed approach.

2.2 MULTILAYER ENCRYPTION: AN APPROACH TO ENHANCE NETWORK SECURITY

Author: A. J. Al-Saraireh, A. A. Al-Ahmad, and A. M. Al-Fayoumi.

Year: 2010

Description

The authors demonstrate the effectiveness of their approach by comparing it to other encryption techniques, such as DES, 3DES, and RSA, and show that their multilayer encryption approach provides a higher level of security. This can be helpful for readers who are interested in learning about different approaches to network security and how multilayer encryption can be used to provide an added layer of protection.

Moreover, the paper discusses the potential impact of the proposed approach on the performance of network communication, and provides experimental results that show that the proposed approach has negligible impact on the performance of the network.

Advantages

The paper proposes a multilayer encryption approach that uses multiple encryption algorithms to enhance the security of network communication. The authors demonstrate the effectiveness of the proposed approach by comparing it to other encryption techniques, such as DES, 3DES, and RSA, and show that their multilayer encryption approach provides a higher level of security.

Disadvantages

The paper does not provide a detailed analysis of the scalability of the proposed multilayer encryption approach. This may be a concern in large-scale network environments where scalability is an important consideration.

2.3 MULTILAYER ENCRYPTION USING ADVANCED ENCRYPTION STANDARD (AES) AND RIVEST-SHAMIR-ADLEMAN (RSA) ALGORITHM

Author: J. Jaiswal, M. K. Soni, and N. K. Shukla.

Year: 2013

Description

The paper proposes a multilayer encryption scheme that uses both Advanced Encryption Standard (AES) and Rivest-Shamir-Adleman (RSA) algorithm to enhance the security of data transmission over the internet. The proposed scheme uses AES encryption to encrypt the data and RSA encryption to encrypt the AES key used for encryption. This creates a double layer of encryption, making it difficult for an attacker to intercept and decrypt the data.

Advantages

- **Increased security:** By using multiple encryption algorithms, the multilayer encryption scheme proposed in this paper provides an additional layer of security to the data. Even if one of the algorithms is compromised, the data remains protected by the other algorithm.
- **Flexibility:** The scheme allows for flexibility in choosing the encryption algorithms based on the level of security required for the data. For instance, AES is known for its high-speed encryption and decryption, whereas RSA is known for its strong encryption key strength.

Disadvantages

Performance: Using multiple encryption algorithms can slow down the encryption and decryption process. In some cases, the additional computational load can cause a significant delay in the processing of large amounts of data.

2.4 MULTI-LAYER ENCRYPTION SCHEME FOR SECURE DATA TRANSMISSION

Author: R. S. Sengar and S. Kumar.

Year: 2015

Description

The multi-layer encryption scheme proposed by R. S. Sengar and S. Kumar in 2015 is a technique for secure data transmission that uses multiple layers of encryption to protect the data from unauthorized access. The scheme involves using two different encryption algorithms, a symmetric key encryption algorithm and an asymmetric key encryption algorithm, in a layered manner. The process starts with the sender encrypting the plaintext using a symmetric key algorithm like Advanced Encryption Standard (AES). The sender generates a random key to encrypt the plaintext, and then encrypts the key using the public key of the receiver, which is obtained from a public key infrastructure (PKI). The encrypted key and ciphertext are then transmitted to the receiver. The receiver decrypts the encrypted key using their private key and then uses the key to decrypt the ciphertext.

Advantages

- The use of multiple layers of encryption provides a higher level of security than using just one encryption algorithm, making it more difficult for an attacker to decrypt the data.
- The symmetric encryption algorithm provides fast encryption and decryption, which can be useful for encrypting large amounts of data.

Disadvantages

The multi-layer encryption scheme can be computationally expensive, as both symmetric and asymmetric encryption algorithms are used. The need to exchange public keys through a PKI can introduce additional complexity and overhead.

2.5 MULTILAYER ENCRYPTION AND STRGANOGRAPHY: AN EFFECTIVE DATA HIDING TECHNIQUE

Author: N. H. Al-Fayoumi, A. J. Al-Saraireh, and A. A. Al-Ahmad.

Year: 2012

Description

The technique involves using a combination of symmetric and asymmetric encryption algorithms and steganography techniques to provide a higher level of security for the hidden data. The process starts with the sender encrypting the plaintext using a symmetric key algorithm like Advanced Encryption Standard (AES). The sender generates a random key to encrypt the plaintext and then uses steganography techniques to embed the encrypted data within an image file. Next, the sender encrypts the symmetric key used to encrypt the plaintext using an asymmetric key algorithm like RSA. The sender then embeds the encrypted symmetric key within the same image file using steganography techniques. The image file with the hidden data is then transmitted to the receiver. The receiver can then use the symmetric key to decrypt the embedded encrypted data within the same image file.

Advantages

The use of multilayer encryption and steganography can provide a high level of security and confidentiality for the data being transmitted. Steganography can be used to hide the encrypted data within other files or data, making it less likely to be detected or targeted by an attacker.

Disadvantages

The use of multiple layers of encryption and steganography can be computationally expensive, particularly for large amounts of data. The security of the scheme is dependent on the strength of the encryption and steganography algorithms used, as well as the secrecy of the keys used.

2.6 TOR: THE SECOND-GENERATION ONION ROUTER

Author: Roger Dingledine, Nick Mathewson, and Paul Syverson.

Year: 2004

Description

Tor network and its onion routing protocol, which allow users to communicate anonymously over the internet. The paper describes the challenges of achieving anonymity online, including the limitations of previous anonymizing systems, and introduces the design and implementation of the Tor network. The authors propose a circuit-based architecture for the Tor network, in which user data is encrypted and routed through a series of relays that are operated by volunteers around the world. Each relay in the circuit decrypts and re-encrypts the data, so that no single relay can link the user to their destination. The authors also describe the use of multiple layers of encryption, or "onion layers," to protect user data as it passes through the network.

Advantages

- The main advantage of the onion routing protocol is that it provides a high degree of anonymity and privacy for users who want to access the internet without revealing their identity or location.
- This can be especially important for individuals who live in countries with repressive regimes, or who want to protect themselves from surveillance by governments, corporations, or other entities.

Disadvantage

One potential disadvantage of the onion routing protocol is that it can slow down internet connections due to the multiple layers of encryption and routing required.

2.7 TOR: DESIGN AND IMPLEMENTATION OF A TOR-BASED ANONYMOUS COMMUNICATION SYSTEM

Author: J. Appelbaum, R. Dingledine, and N. Mathewson.

Year: 2012

Description

The paper describes the design and implementation of the Tor network, including the circuit-based architecture and the use of onion layers to protect user data. The authors discuss the challenges of achieving anonymity in the face of adversaries who can monitor or manipulate network traffic, and describe the defenses built into the Tor network to protect against such attacks. These include techniques for preventing traffic analysis, or the identification of users based on the patterns of their network activity.

Advantages

- The paper provides a detailed technical overview of the Tor network and its onion routing protocol, highlighting its strengths and the challenges faced in achieving anonymity online.
- The paper also describes the defenses built into the Tor network to protect against various forms of attacks, such as traffic analysis, and provides information on the Tor Browser Bundle, a tool that allows users to access the network and browse the web anonymously.

Disadvantages

- One potential disadvantage of the Tor network discussed in the paper is that it can be slow due to the multiple layers of encryption and routing required to maintain anonymity.
- Additionally, while the Tor network provides a high degree of anonymity, it is not fool proof, and there are still ways for determined adversaries to potentially identify or track users.

2.8 ONION ROUTING FOR ANONYMOUS AND PRIVATE INTERNET CONNECTIONS

Author: Aaron Johnson, Paul Syverson, and Roger Dingledine.

Year: 2006

Description

The paper presents an overview of the onion routing protocol, including its design and implementation, and analyzes its security properties in the context of various attacks and threat models. The authors describe the use of onion layers to protect user data as it passes through the network, and analyze the strengths and weaknesses of this approach in terms of providing anonymity and privacy. They also discuss potential attacks on the protocol, such as traffic analysis and denial of service attacks, and propose defenses to protect against these threats.

Advantages

- The paper provides an early survey of onion routing's security, which is the underlying protocol used in the Tor network, providing an overview of its design and implementation, and analyzing its security properties in the context of various attacks and threat models.
- The paper also proposes defenses to protect against potential attacks on the protocol, such as traffic analysis and denial of service attacks, and identifies open research questions in the field of onion routing security.

Disadvantages

The paper focuses primarily on the technical aspects of onion routing security and may not provide a broader perspective on the ethical and societal implications of online anonymity and privacy.

2.9 THE TOR NETWORK: ANONYMITY AND PRIVACY ON THE INTERNET

Author: Claudia Diaz and Stefaan Seys.

Year: 2015

Description

The paper describes the design and implementation of the Tor network, including its circuit-based architecture and the use of onion layers to protect user data. The authors discuss the importance of online anonymity and privacy in the context of censorship and surveillance, and describe the role of the Tor network in providing a means for users to access the internet anonymously and securely. They also discuss the challenges of achieving anonymity in the face of adversaries who can monitor or manipulate network traffic, and describe the defenses built into the Tor network to protect against such attacks. The paper also describes the Tor directory, which is used to coordinate the operation of the relays that make up the Tor network, as well as the Tor Browser Bundle, a software package that allows users to access the Tor network and browse the internet anonymously.

Advantages

This paper is one of the seminal works on the Tor network and its onion routing protocol, providing a detailed overview of its design and implementation, as well as the challenges and limitations of achieving online anonymity and privacy.

Disadvantages

The paper primarily focuses on the technical aspects of the Tor network and onion routing protocol, and may not provide a broader perspective on the ethical and societal implications of online anonymity and privacy.

2.10 THE DARK SIDE OF THE ONION: EXAMINING THE HARMFUL SIDE EFFECTS OF TOR'S HIDDEN SERVICES

Author: Damon McCoy, Kirill Levchenko, and Geoffrey M. Voelker.

Year: 2012

Description

The authors use network measurement techniques to analyze the growth and changes in the Tor network, including the number and distribution of relays and the characteristics of the network's traffic. The paper describes the methods used to collect and analyze network data, including the use of customized software tools to identify and classify Tor relays and traffic. The authors also present their findings on the growth and changes in the Tor network, including the increase in the number of relays and the distribution of these relays around the world. The paper also discusses the potential implications of these findings for the security and privacy of the Tor network, including the potential for centralized control of the network and the risks associated with running relays in countries with weak legal protections for privacy and free speech.

Advantages

This paper provides a longitudinal study of the growth and changes in the Tor network over a period of several years, using network measurement techniques to analyze the number and distribution of relays and the characteristics of the network's traffic

Disadvantages

One potential disadvantage of this paper is that it focuses primarily on the technical aspects of the Tor network and may not provide a broader perspective on the ethical and societal implications of online anonymity and privacy.

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

In the EXISTING SYSTEM, Most of the Times, Data are not properly transferred to the Destination, as there are more chances for Attacks from the Intermediate Hops. In any Multi Hop Network system, Intermediate Node's Support is mandatory. They should be Reliable for the effective data transfer. But in most of the cases, in the existing system, it is not so. Either Intermediate itself will drop the packets the packets or it would be hacked. We need to deploy IP Tracking system to track the misbehavior of any intermediate Nodes. It is Time consuming, not reliable and finally increasing the Network Traffic.

3.1.1 Disadvantages of Existing System

- Waiting time is increased
- Unreliable
- Packet Loss
- Less effective
- Less security

3.2 PROPOSED SYSTEM

Multilayer encryption is a technique that involves encrypting data multiple times using different encryption algorithms or keys to enhance security. This technique is widely used in secure communication systems and data storage systems to protect sensitive information.

In Our Model, We deploy Multi Layer Secured protocol. Every node while registering, server will be provided with Id, primary key, secondary key and decryption key. Data's first encrypted using RSA algorithm and then with corresponding Master key of all the hops respectively.

Intermediate Node's ID is also Hashed for verification. In the Proposed system, assuming we identify "C" is identified as the best Intermediate node

that connects “A” the Source Node and “E” the Destination Node, first Data is encrypted using RSA Algorithm, then the same Data is encrypted using Master Key of Destination Node “E”, then again the same encrypted data is again Encrypted by the Master Key of Intermediate Node “C”.

Now this triple Encrypted data is transferred to the Intermediate Node “C”. Now the “C” Node’s ID is verified and then its Secret Key is verified. After verification First layer of the data is decrypted. Then it is forwarded to the destination Node “E”. Now second layer is decrypted by verifying the Secret Key of “E”. Now finally the single layer is decrypted using “E” Node’s Decryption Key. Finally Original data is restored to the Destination Node “E”.

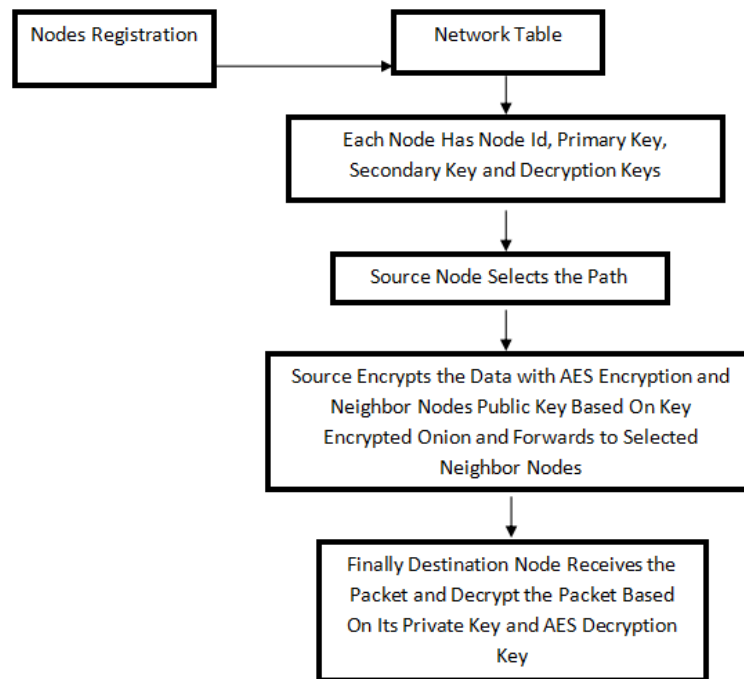


Fig 3.1 Flow diagram of proposed system

3.2.1 Advantages of Proposed System

- Data is transferred securely to the Destination Node
- Most Reliable
- Assured data Transfer which ensures less time for Transfer.
- Best Intermediate Node is identified Using Capacity Calculation

REQUIREMENT SPECIFICATION

4.1 HARDWARE REQUIREMENTS

- Processor : Core i3/i5/i7
- RAM : 2-4GB
- HDD : 500 GB

4.2 SOFTWARE REQUIREMENTS

- Operating System : Windows 7/8/10
- Front End : Java-JDK1.6
- Java Packages : Java Server Page, Swing
- Back End : MYSQL
- IDE : NetBeans 8.1 / 8.2
- Tool : Apache Tom Cat 5.5.9

4.3 REQUIREMENT ANALYSIS

Requirement analysis, also called requirement engineering, is the process of determining user expectations for a new modified product. It encompasses the tasks that determine the need for analysing, documenting, validating and managing software or system requirements. The requirements should be documentable, actionable, measurable, testable and traceable related to identified business needs or opportunities and define to a level of detail, sufficient for system design.

4.3.1 Functional Requirements

Functional requirements in software engineering refer to the specific actions that a software system or application must perform in order to meet the needs of its users. These requirements are typically described in detail in a software requirements specification document, which outlines the functionality

and features that are expected from the software. Functional requirements can include a wide range of different features and capabilities, depending on the specific needs of the users and the intended use of the software.

Encryption Algorithm

The system must support multiple encryption algorithms, including AES, RSA, Blowfish, etc. These algorithms ensure that data is protected from unauthorized access.

Key Management

The system must have a secure key management mechanism that generates, stores, and exchanges encryption keys. It should also provide a secure way of exchanging keys between different systems.

Data Integrity

The system must ensure the integrity of data by implementing checksums or message authentication codes (MACs). This ensures that the data is not modified or tampered with during transmission.

Access Control

The system must have access control mechanisms to authenticate and authorize users. This ensures that only authorized users can access the data.

Decryption

The system must allow authorized users to decrypt the encrypted data. It should also provide a mechanism for revoking access when necessary.

Cross-Platform Compatibility

The system must be compatible with different platforms and operating systems to ensure seamless data exchange between them.

Scalability

The system must be scalable to accommodate growing data volumes. It should be able to handle large amounts of data without compromising performance.

4.3.2 Non-Functional Requirements

Security

The system must have high-security measures to prevent unauthorized access to data. It should include mechanisms such as firewalls, intrusion detection, and prevention systems, etc.

Performance

The system must perform encryption and decryption processes within acceptable time limits. The system should not slow down or cause delays in data transmission.

Reliability

The system must be reliable to ensure data integrity and prevent data loss. The system should be designed to handle failures without losing data.

Usability

The system must be user-friendly, with easy-to-use interfaces for users to interact with. Users should be able to access the system and perform their tasks without requiring technical expertise.

Compatibility

The system must be compatible with different hardware and software configurations. It should be able to integrate with other systems seamlessly.

Availability

The system must be available 24/7 to ensure data accessibility. The system should have mechanisms such as redundancy and failover to ensure high availability.

Maintainability

The system must be easy to maintain, update, and upgrade. The system should have mechanisms for easy troubleshooting and bug fixing.

SYSTEM DEVELOPMENT

5.1 SYSTEM ARCHITECTURE

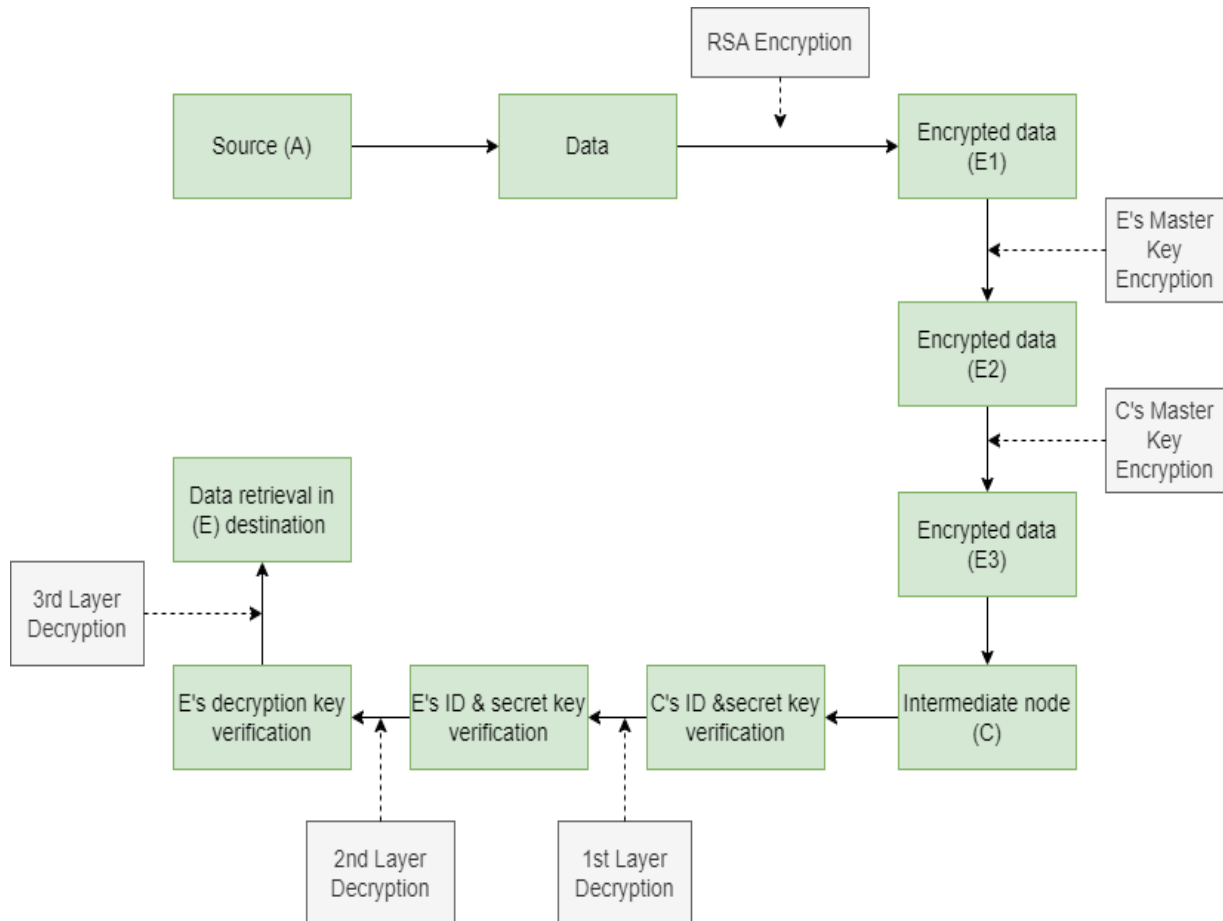


Fig 5.1 System Architecture

A system architecture is the conceptual model that defines the structure, behavior, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system.

In Our Model, We deploy Multi Layer Secured protocol. Every node while registering, server will be provided with Id, primary key, secondary key and decryption key. Data's first encrypted using RSA algorithm and then with corresponding Master key of all the hops respectively.

Intermediate Node's ID is also Hashed for verification. In the Proposed system, assuming we identify "C" is identified as the best Intermediate node that connects "A" the Source Node and "E" the Destination Node, first Data is encrypted using RSA Algorithm, then the same Data is encrypted using Master Key of Destination Node "E", then again the same encrypted data is again Encrypted by the Master Key of Intermediate Node "C".

Now this triple Encrypted data is transferred to the Intermediate Node "C". Now the "C" Node's ID is verified and then its Secret Key is verified. After verification First layer of the data is decrypted. Then it is forwarded to the destination Node "E". Now second layer is decrypted by verifying the Secret Key of "E". Now finally the single layer is decrypted using "E" Node's Decryption Key. Finally Original data is restored to the Destination Node "E".

Here is a detailed system architecture for implementing multilayer encryption

Input Data

The first step in the multilayer encryption system is to input the data that needs to be encrypted. The input data can be any type of digital data, such as text.

Data Encryption

The third step is to encrypt the input data using the selected encryption algorithms. Each encryption algorithm will be applied to the input data in a separate layer. The output of each encryption layer will be the input to the next layer.

Key Management

The fourth step is to manage the keys that are used for each encryption

layer. Each layer may have its own set of keys, and these keys must be managed securely to ensure that the encrypted data cannot be decrypted without authorization.

Data Storage

The fifth step is to store the encrypted data in a secure location. The data storage system should be designed to protect the encrypted data from unauthorized access and ensure that the data can be retrieved when needed.

Data Retrieval

The sixth step is to retrieve the encrypted data when needed. The data retrieval process should ensure that the encrypted data is decrypted in the correct order, using the correct keys, and with the correct encryption algorithms.

Decryption

The final step is to decrypt the encrypted data. The decryption process should be designed to ensure that the decrypted data is accurate and that it can be used for its intended purpose.

5.2 UML DIAGRAM

Unified Modeling Language (UML) is a standard graphical notation for describing and visualizing software systems. UML diagrams are used to model and design software systems, and they provide a common language for software developers to communicate their ideas and designs to each other. There are several types of UML diagrams, each of which is used to represent a different aspect of the system being designed.

Advantages

- To represent complete systems (instead of only the software portion) using object oriented concepts
- To establish an explicit coupling between concepts and executable code
- To take into account the scaling factors that are inherent to complex and critical systems

5.2.1 USE CASE DIAGRAM

Use case diagrams overview the usage requirement for system. they are useful for presentations to management and/or project stakeholders, but for actual development you will find that use cases provide significantly more value because they describe “the meant” of the actual requirements. A use case describes a sequence of action that provide something of measurable value to an action and is drawn as a horizontal ellipse.

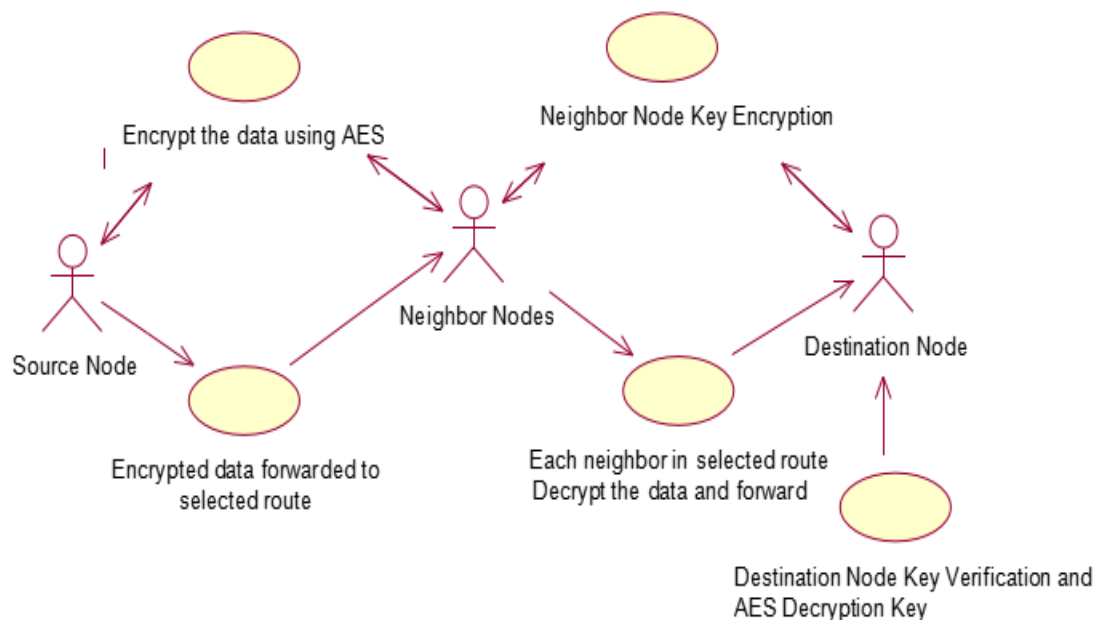


Fig 5.2 Use Case Diagram

5.2.2 CLASS DIAGRAM

A class diagram is a type of UML (Unified Modeling Language) diagram that is used to represent the structure of a system in terms of its classes, attributes, and methods. It is a static diagram that provides a high-level overview of a system's architecture and serves as a blueprint for designing and implementing software systems.

A class diagram consists of three main components: classes, associations, and multiplicities. Classes are represented as rectangles with the class name inside. The class attributes are listed below the class name, while the class methods are listed below the attributes.

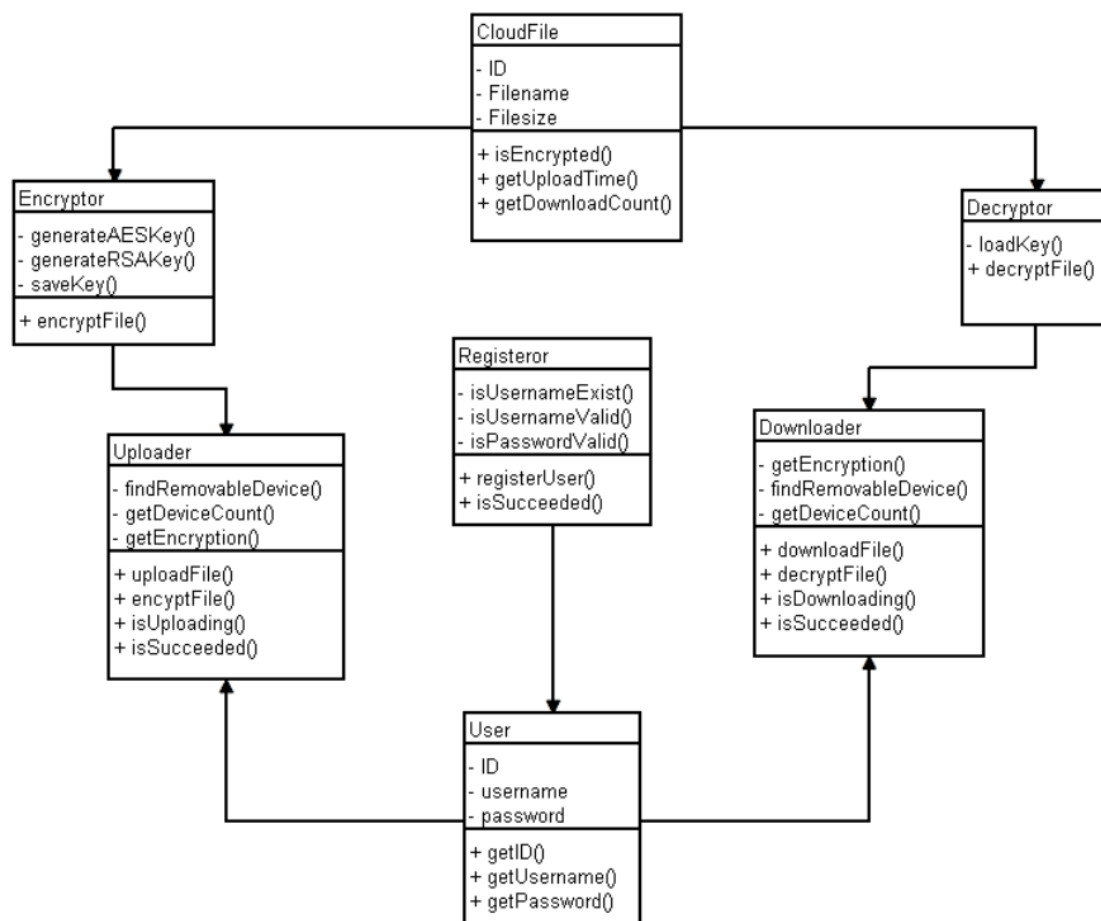


Fig 5.3 Class Diagram

5.2.3 SEQUENCE DIAGRAM

Sequence diagram model the flow of logic within your system in a visual manner, enabling you both to document and validate your logic, and commonly used for both analysis and design purpose. Sequence diagram are the most popular UML artifact for dynamic modeling, which focuses on identifying the behavior within your system.

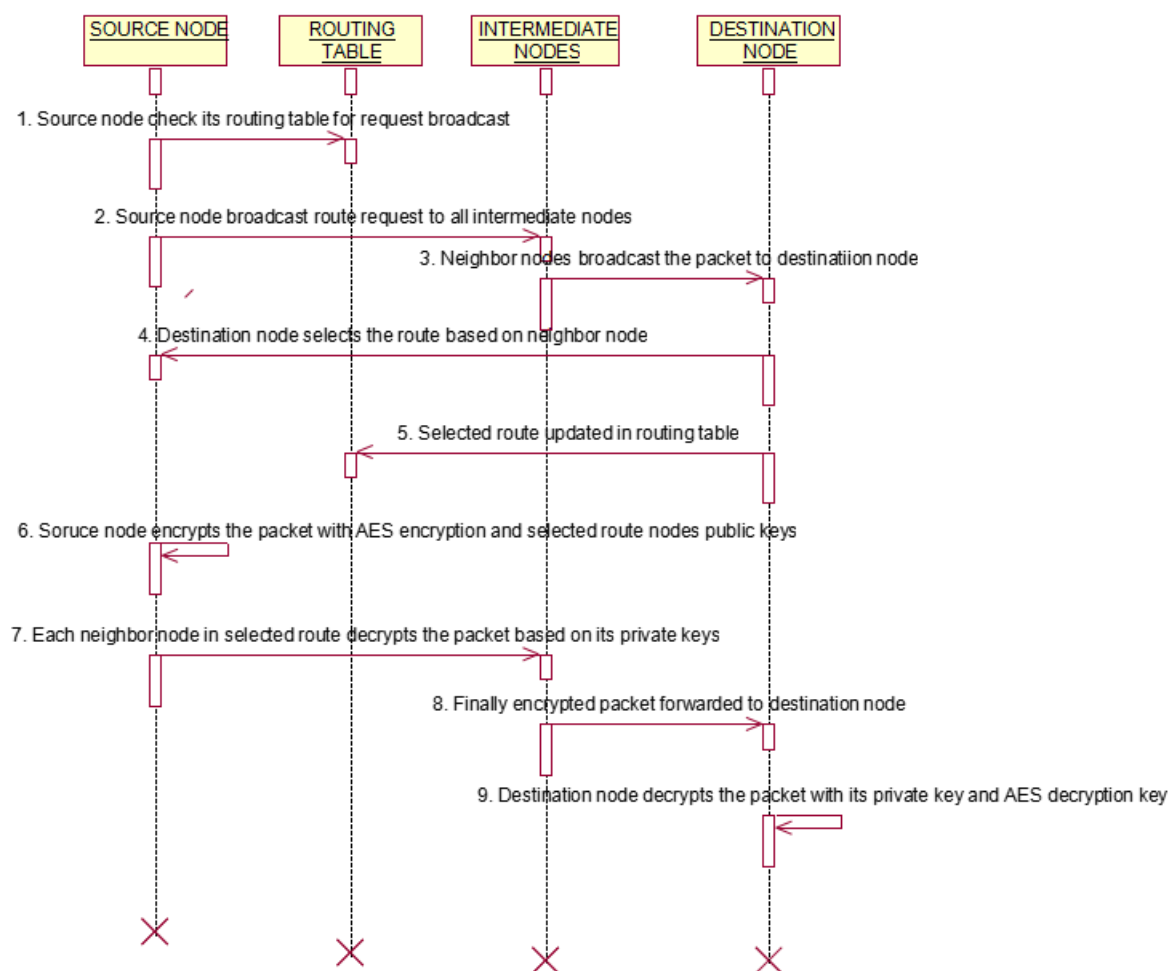


Fig 5.4 Sequence Diagram

5.2.4 COLLABORATION DIAGRAM

A collaboration diagram, also known as a communication diagram, is a type of UML (Unified Modeling Language) diagram that is used to visualize the interactions between objects or roles in a system. It shows how different objects or roles collaborate to achieve a specific task or goal.

In a collaboration diagram, objects are represented as rectangles with the object name inside. Roles are represented as stick figures with the role name inside. The interactions between objects and roles are shown as arrows between them. The arrows indicate the direction of the communication and can be labeled with messages or actions that are exchanged between the objects and roles.

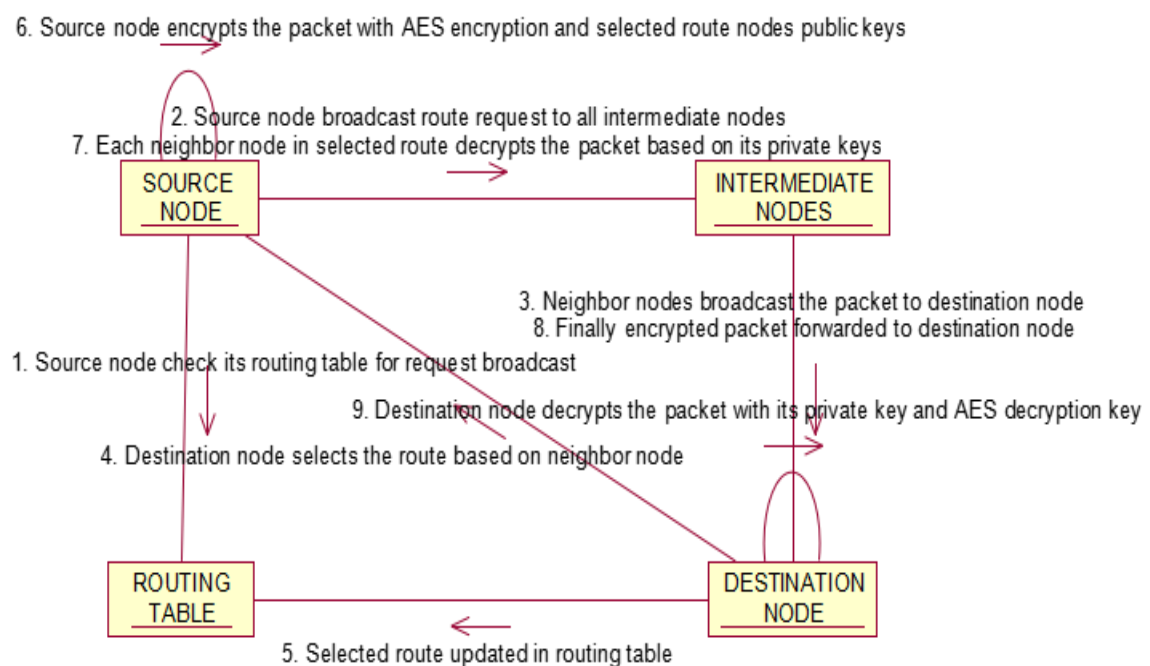


Fig 5.5 Collaboration Diagram

5.2.5 ACTIVITY DIAGRAM

Activity diagram are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. The activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. Activity diagram consist of Initial node, activity final node and activities in between.

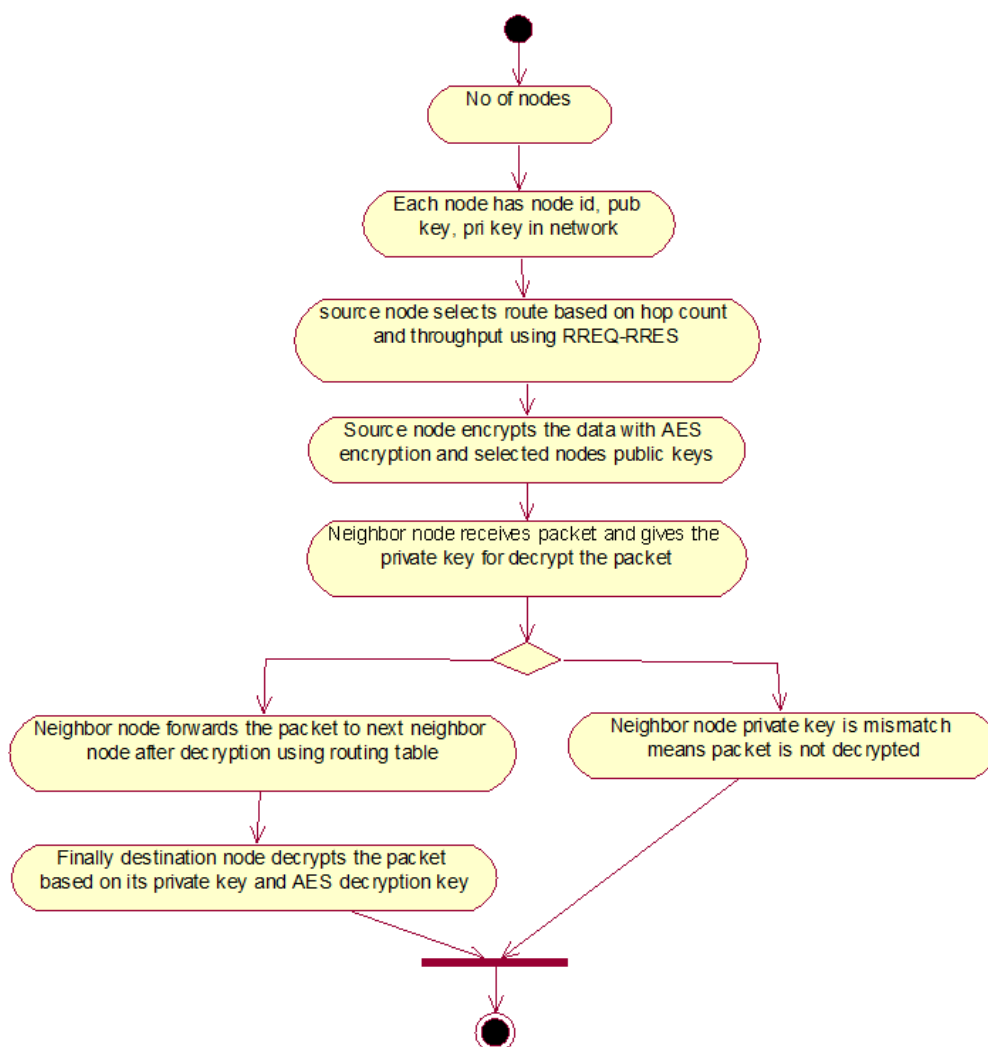


Fig 5.6 Activity Diagram

5.2.6 ENTITY RELATIONSHIP DIAGRAM

ER (Entity-Relationship) diagrams are visual representations used to model the structure of a database. They are widely used in database design to illustrate the relationships between entities and their attributes.

An entity is a real-world object or concept that has a distinct existence and can be identified. In a database, entities are represented by tables, and each row in a table represents a specific instance of that entity. An attribute is a property or characteristic of an entity, and it is represented by a column in a table. An ER diagram consists of three main components: entities, relationships, and attributes.

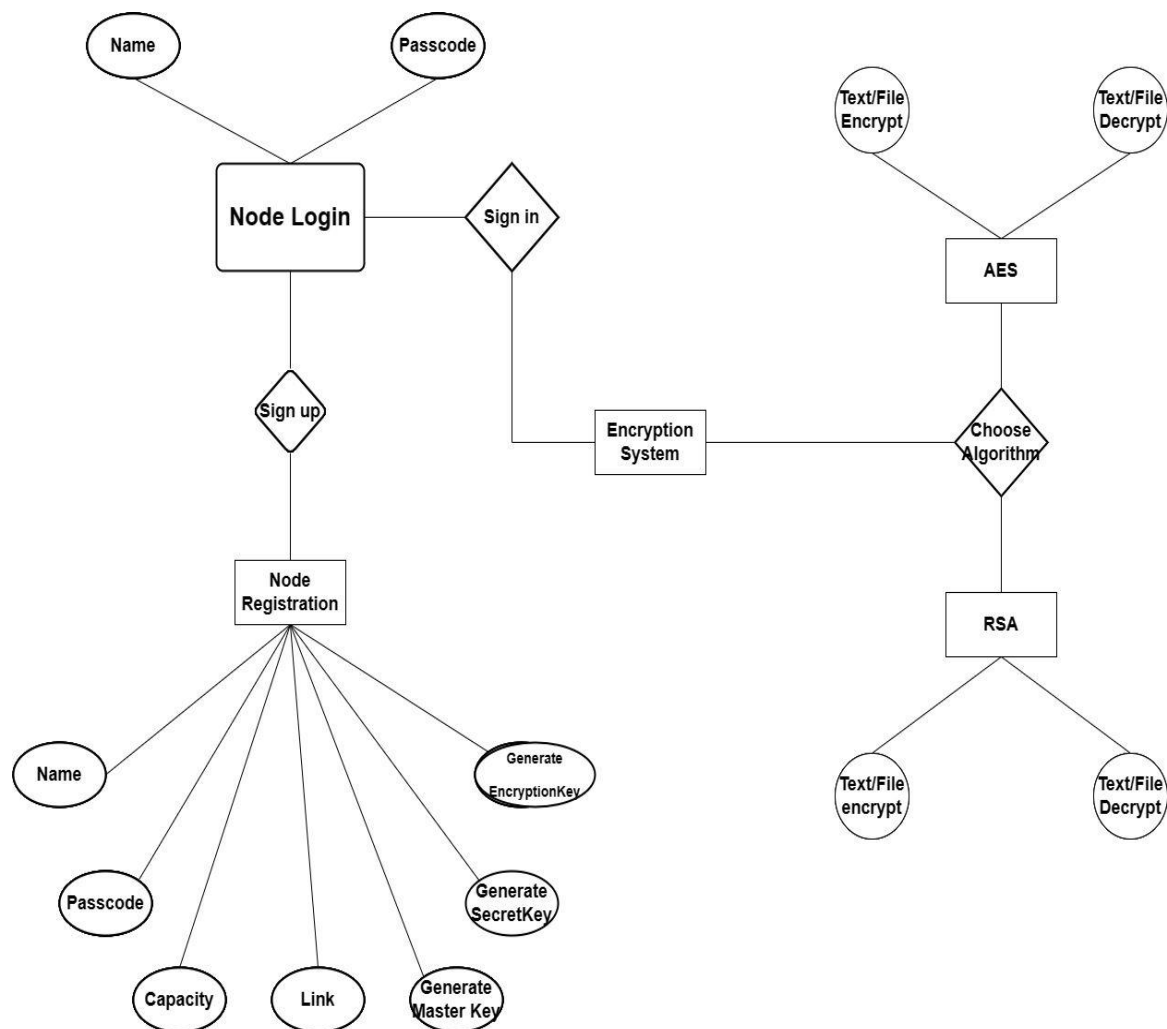


Fig 5.7 Entity Relationship Diagram

5.2.7 DATA FLOW DIAGRAM

LEVEL 0

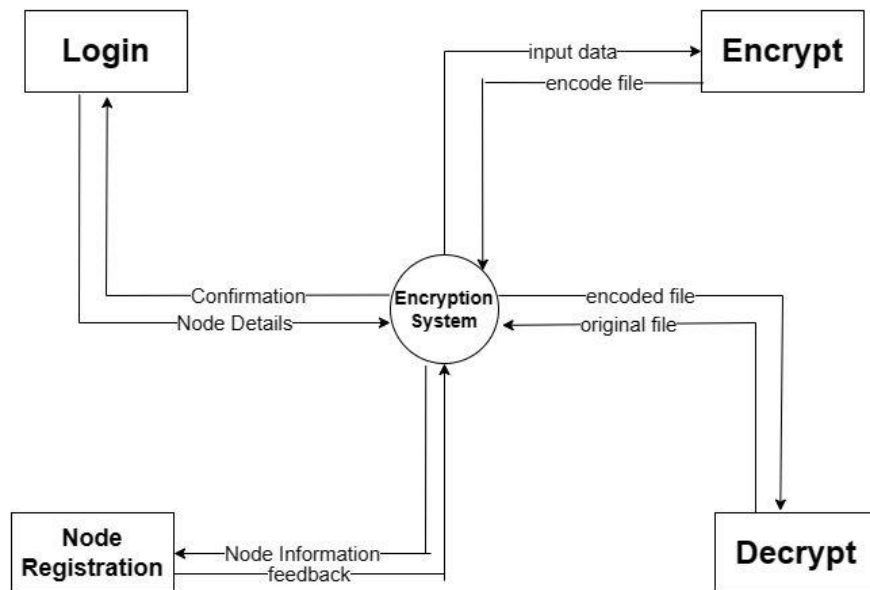


Fig 5.8 Level 0 DFD Diagram

LEVEL 1

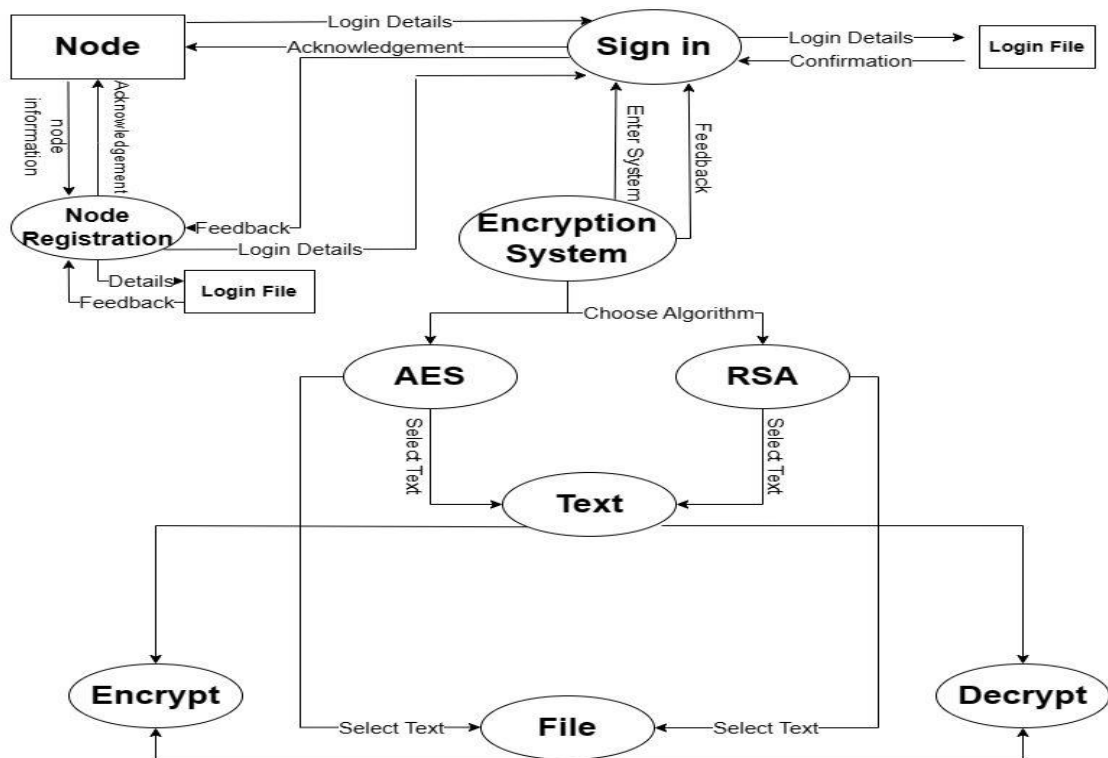


Fig 5.9 Level 1 DFD Diagram

5.3 FEASIBILITY STUDY

Social Feasibility

Multi-layer encryption is becoming increasingly important due to the growing threat of cyber-attacks and data breaches. Therefore, the acceptance and adoption of this technology are quite high. Additionally, there are privacy and data protection laws in place that require organizations to implement measures to safeguard sensitive data. As a result, implementing a multi-layer encryption project is socially feasible, provided that the project complies with legal and regulatory requirements and meets the needs of the users.

Technical Feasibility

Multi-layer encryption involves adding multiple layers of security to protect data. This can be achieved through various encryption methods, including symmetric key encryption, asymmetric key encryption, and hashing. The technical feasibility of a multi-layer encryption project depends on the availability of necessary resources, compatibility with existing infrastructure, and the complexity of the encryption system. If the necessary resources are available, and the encryption system can be integrated with the existing infrastructure, the project is technically feasible.

Economic Feasibility

Implementing a multi-layer encryption project involves costs associated with acquiring the necessary resources, including hardware, software, and personnel. However, the cost of a data breach can be much higher, which includes reputational damage, legal costs, and loss of revenue. Therefore, implementing a multi-layer encryption project can be economically feasible if it helps to prevent costly data breaches. Additionally, the project can also lead to a competitive advantage by demonstrating a commitment to data security, which can attract more customers and increase revenue.

MODULES

A modular design reduces complexity, facilitates change (a critical aspect of software maintainability), and results in easier implementation by encouraging parallel development of different parts of the system. Software with effective modularity is easier to develop because functions may be compartmentalized and interfaces are simplified. Software architecture embodies modularity that is software is divided into separately named and addressable components called modules that are integrated to satisfy problem requirements.

Modularity is the single attribute of software that allows a program to be intellectually manageable. The five important criteria that enable us to evaluate a design method with respect to its ability to define an effective modular design are: Modular decomposability, Modular Comprehensibility, Modular Understandability, Modular continuity, Modular Protection. The following are the modules of the project, which is planned in aid to complete the project with respect to the proposed system, while overcoming existing system and also providing the support for the future enhancement.

The modules are

- Network construction
- First level encryption
- Multi layer encryption model
- Decryption Process

6.1 NETWORK CONSTRUCTION

In this Project concept, first we have to construct a network which consists of 'n' number of Nodes. So that nodes can request data from other nodes in the network. Since the Nodes have the mobility property, we can assume that the nodes are moving across the network. Network is used to store all the Nodes information like Node Id and other information. Each node is

having Master Key & secondary key. Also network will monitor all the Nodes Communication for security purpose.

The construction of a network refers to the design and implementation of the physical and logical components that make up a computer network. A network consists of multiple devices, such as computers, servers, routers, switches, and other networking equipment, that are interconnected to enable the exchange of data and communication between them. The network construction process typically involves several components, including:

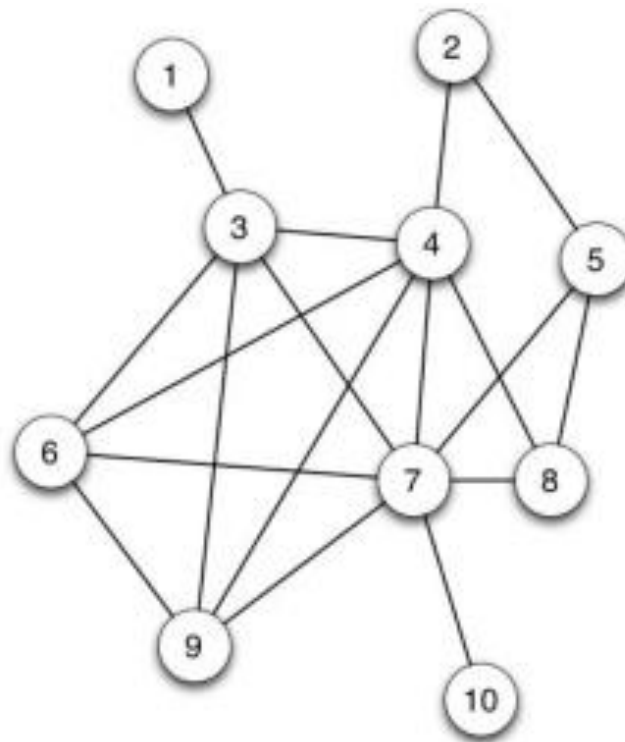


Fig 6.1 Network Construction

Network Topology

This refers to the physical or logical layout of the network. There are several network topologies, including star, bus, ring, mesh, and hybrid. The choice of network topology depends on the size and complexity of the network, as well as the requirements for performance, reliability, and security.

Network Architecture

This refers to the overall design of the network, including the protocols, standards, and technologies used to enable communication between the devices. The network architecture includes the network layers, such as the application layer, transport layer, network layer, data link layer, and physical layer.

Network Devices

These are the physical components that make up the network, such as routers, switches, hubs, repeaters, and firewalls. These devices are used to connect the devices on the network and enable communication between them.

Network Security

This refers to the measures taken to protect the network from unauthorized access, hacking, and other security threats. Network security includes the use of firewalls, intrusion detection systems, encryption, and other security mechanisms.

The network construction process involves planning, designing, and implementing the network infrastructure, as well as configuring and testing the network components to ensure that they function properly. It is important to consider the scalability, performance, reliability, and security of the network when constructing a network, as well as the needs of the users and applications that will be using the network.

6.2 FIRST LEVEL ENCRYPTION

In this module the data which is to be transmitted from the source is first encrypted using RSA algorithm before sending it to the destination. The data security is ensured using this module. This encryption process is used as the initial Security process followed by multi security Encryption Process. Encryption of data is the process of converting plaintext into ciphertext to protect the confidentiality and integrity of the data. Encryption is an essential security mechanism that is widely used to protect sensitive information, such as personal data, financial information, and classified information, from

unauthorized access and theft. In this article, we will discuss the encryption of data in detail, including the types of encryption, encryption algorithms, and the importance of encryption in data security.

Types of Encryption

There are two main types of encryption

- Symmetric encryption
- Asymmetric encryption.

Symmetric encryption, also known as shared secret encryption, uses the same key for both encryption and decryption. The key is shared between the sender and the receiver, and it is used to encrypt and decrypt the data. Examples of symmetric encryption algorithms include Advanced Encryption Standard (AES), Data Encryption Standard (DES), and Triple DES (3DES).

Asymmetric encryption, also known as public key encryption, uses two different keys for encryption and decryption. One key is a public key, which is widely distributed and can be used to encrypt the data. The other key is a private key, which is kept secret by the owner and is used to decrypt the data. Examples of asymmetric encryption algorithms include Rivest-Shamir-Adleman (RSA), Elliptic Curve Cryptography (ECC), and Digital Signature Algorithm (DSA).

RSA is an asymmetric encryption algorithm that is widely used for encrypting data and creating digital signatures.

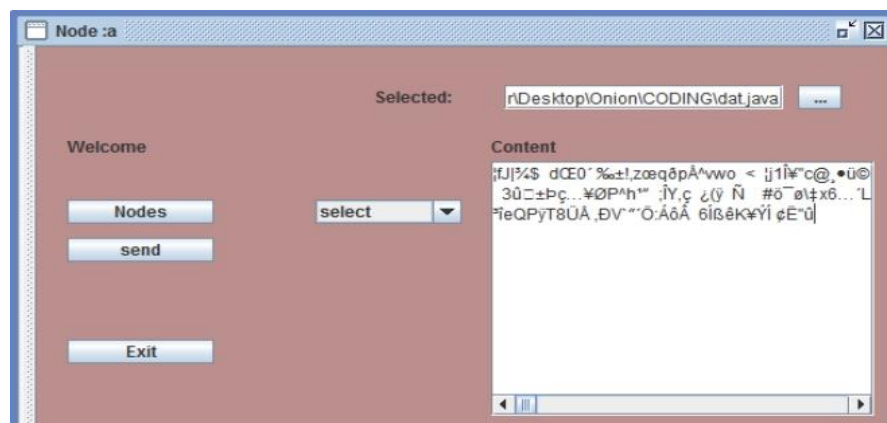


Fig 6.2 First Layer Encryption

6.3 MULTI LAYER ENCRYPTION MODEL

In this Module, the main Project implementation of Multi Layer Security is achieved. After first level of Encryption, the same encrypted data is again encrypted by the Master Key of Destination Node and one another Encryption using Intermediate Node. Now this triple Encrypted data is transferred to the Intermediate Node. Now our System will verify the Intermediate Node's ID and then its Secret Key. After verification First layer of the data is decrypted. Then it is forwarded to the destination Node. Now second layer is decrypted by verifying the Secret Key of Destination Node. Now finally the single layer is decrypted using Destination's Decryption Key. Finally Original data is restored to the Destination Node.

In this module two algorithms are used

- RSA algorithm
- SHA-256 algorithm

RSA algorithm

RSA (Rivest-Shamir-Adleman) is an asymmetric encryption algorithm that is widely used for encrypting data and creating digital signatures. It was invented in 1977 by Ron Rivest, Adi Shamir, and Leonard Adleman, and is named after their surnames. RSA is based on the mathematical principles of number theory and is considered one of the most secure encryption algorithms available.

RSA is a public-key encryption algorithm, which means that it uses two different keys for encryption and decryption. One key is a public key, which is widely distributed and can be used to encrypt the data. The other key is a private key, which is kept secret by the owner and is used to decrypt the data. The strength of RSA lies in the difficulty of factoring large numbers into their prime factors, which is the basis for generating the public and private keys.

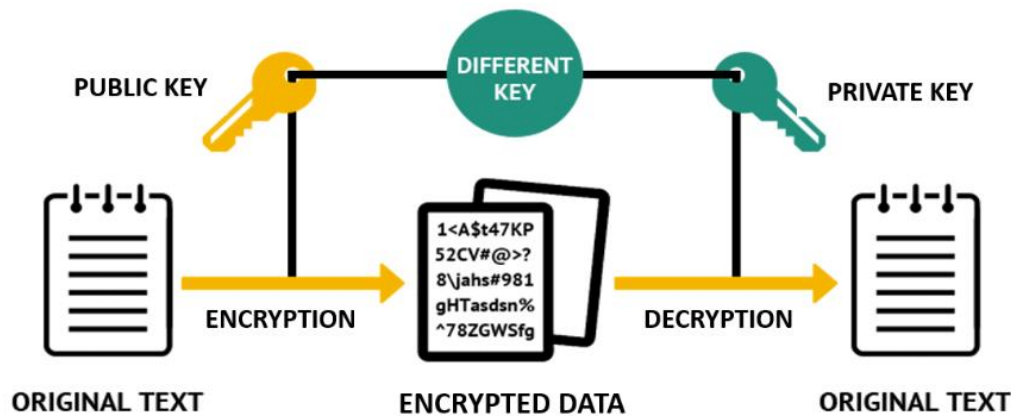


Fig 6.3 RSA Algorithm

The process of generating RSA keys involves the following steps

- Select two prime numbers, p and q , where $p \neq q$. These prime numbers are used to generate the public and private keys.
- Calculate the product of the two prime numbers, $n = p * q$. This value is used as the modulus for the keys.
- Calculate Euler's totient function of n , $\phi(n) = (p-1) * (q-1)$. This value is used to determine the public and private keys.
- Select an integer e , where $1 < e < \phi(n)$, and e is co-prime with $\phi(n)$. This value is used as the public key.
- Calculate the value of d , where $d * e \equiv 1 \pmod{\phi(n)}$. This value is used as the private key.

Once the keys are generated, the encryption and decryption process using RSA can be explained as follows

Encryption

- The plaintext message is converted into a number, m , using a predetermined method, such as ASCII or Unicode.
- The public key (n, e) is used to encrypt the message, using the formula $c = m^e \pmod{n}$.

- The encrypted message, c , is sent to the receiver.

Decryption

- The receiver uses their private key, d , to decrypt the message, using the formula $m = c^d \pmod{n}$.
- The decrypted message, m , is converted back into its original format, such as ASCII or Unicode, to reveal the plaintext message.

RSA is widely used in various applications, such as secure email, digital signatures, and secure online transactions. It is considered one of the most secure encryption algorithms available, with key sizes ranging from 1024 bits to 4096 bits. However, RSA is not without its limitations, as it can be vulnerable to certain attacks, such as timing attacks and side-channel attacks, if not implemented properly.

SHA-256 algorithm

SHA-256 (Secure Hash Algorithm 256-bit) is a cryptographic hash function that generates a fixed-length output of 256 bits, regardless of the size of the input data. It was developed by the National Security Agency (NSA) of the United States and is widely used for digital signatures, password authentication, and other applications that require data integrity.

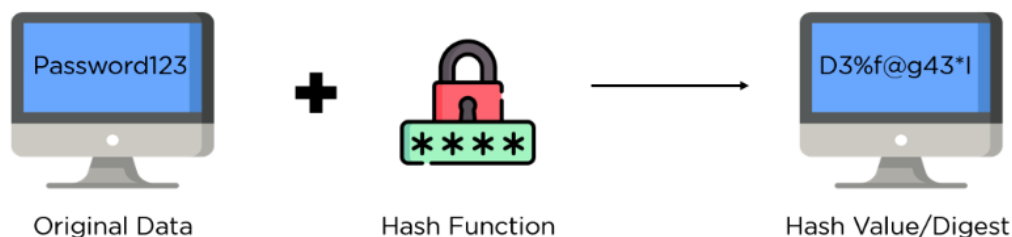


Fig 6.4 SHA-256 Algorithm

The SHA-256 algorithm works by taking an input message of any length and processing it in 512-bit blocks. The input message is first padded to a multiple of 512 bits, and then broken into blocks of 512 bits each. Each block is then processed using a series of logical operations to produce a fixed-length output of 256 bits. The steps involved in the SHA-256 algorithm can be summarized as follows

- **Padding:** The input message is padded with zeros to ensure that its length is a multiple of 512 bits.
- **Message Schedule:** The padded message is divided into 512-bit blocks, and each block is further divided into 16 32-bit words.
- **Initial Hash Values:** The algorithm uses eight 32-bit values as initial hash values, which are obtained by taking the first 32 bits of the fractional parts of the square roots of the first eight prime numbers.
- **Compression Function:** The compression function is applied to each 512-bit block of the padded message and the initial hash values. The function consists of several rounds, each of which performs a series of logical operations to transform the input values into output values.
- **Output:** The output of the compression function is a 256-bit message digest that represents the input message.

The SHA-256 algorithm is designed to be resistant to both preimage attacks and collision attacks. A preimage attack is an attempt to find a message that produces a given hash value, while a collision attack is an attempt to find two different messages that produce the same hash value. The 256-bit output of SHA-256 provides a high level of security, as the probability of a collision is extremely low.

6.4 DECRYPTION PROCESS

In this module, neighbor node decrypts the packet and finally sends to destination node. Then the destination node decrypts the packet with its private key and AES decryption key. Finally destination node views the original data.

Since the paths capacity will vary dynamically, so that the paths will be changed dynamically as per data transfer along the network. So it increases the packet delivery ratio and decreases the average end-to-end delay.

Decryption is the process of converting encrypted or encoded data back into its original form. It involves using a decryption key, which is a special code or password, to unlock the encrypted data and make it readable again.

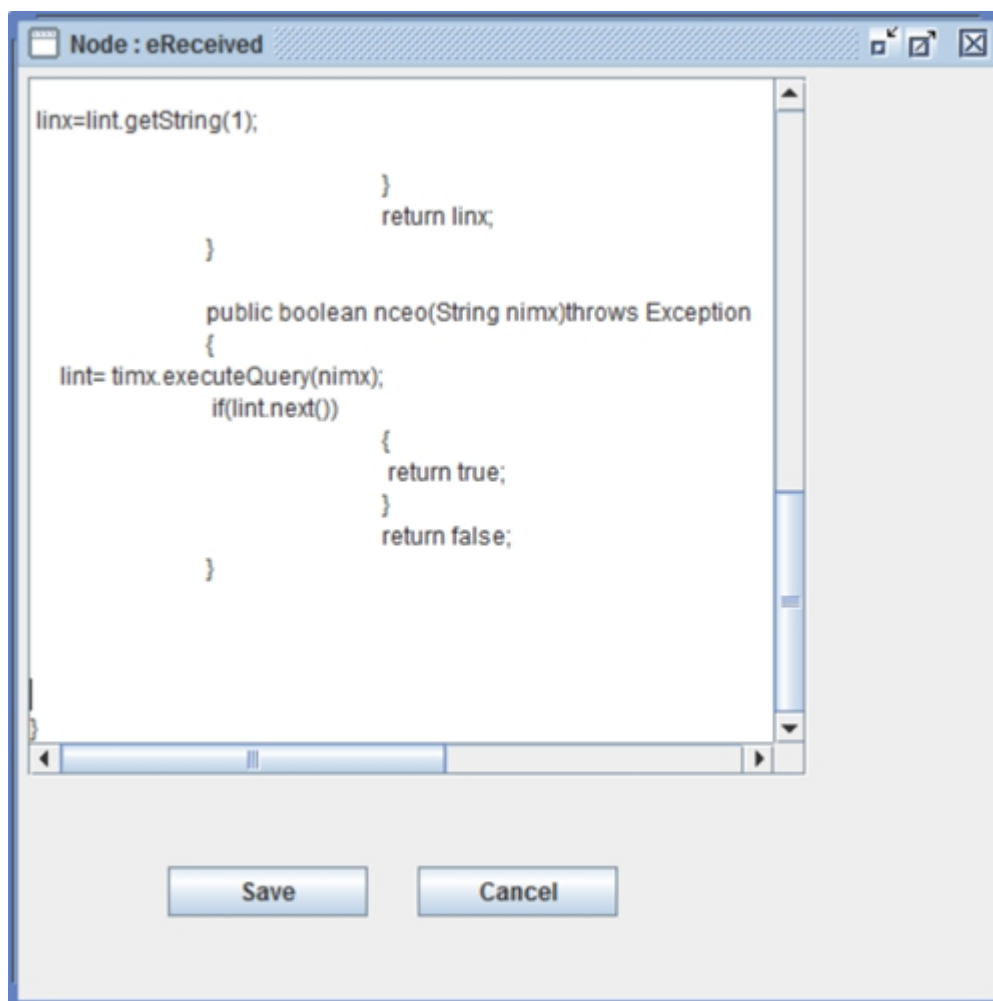


Fig 6.5 Data Decryption

IMPLEMENTATION

7.1 SAMPLE CODE

Node Creation

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.net.*;
import java.util.*;
public class ine extends JFrame
{
String intf=null,lemt=null;
public JButton neck;
public JLabel intl;
public JLabel clon;
public JTextField inte;
public JTextField fldm;
public JTextField asount;
public JLabel ulem;
public JLabel clem;
public JPasswordField lcom;
public JButton lotm;
public JButton clef;
public JButton cent;
public JTextField lety;
public JPanel lint;
public JPanel jlem;
public ine(int ineo) throws Exception
{
```

```

super();
fint=new method();
ndetail();
fint.imet(String.valueOf(ineo));
initializeComponent();
this.setVisible(true);
}
public void initializeComponent()
{
intl=new JLabel("Email");
clon=new JLabel("Mobile");

inte=new JTextField();
fldm=new JTextField();

neck=new JButton("Back");
ulem = new JLabel("Name");
ulem.setForeground(Color.blue);
clem = new JLabel("Passcode");
clem.setForeground(Color.blue);
lety = new JTextField("");
lcom = new JPasswordField();
lotm = new JButton("Login");
clef = new JButton("Sign");
cent=new JButton("exit");
lint = (JPanel)this.getContentPane();
jlem = new JPanel();
lcox = new JButton("Submit");
clex = new JButton("Clean");

```

```

nimx = new JButton("Connection");
ulex = new JLabel("Name");
nlex = new JLabel("Link");
nlex.setForeground(Color.blue);
cldm = new JLabel("Passcode");
maot=new JLabel("Amount");
maot.setForeground(Color.blue);
asount=new JTextField();
letw = new JTextField();
letm = new JPasswordField();
ulex.setForeground(Color.blue);
cldm.setForeground(Color.blue);
intl.setForeground(Color.blue);
clon.setForeground(Color.blue);
neoy=new JComboBox(neox);
lotm.addActionListener(new ActionListener() {
public void actionPerformed(ActionEvent eint)
{
logm_actionPerformed(eint);
}
});
neck.addActionListener(new ActionListener() {
public void actionPerformed(ActionEvent eint)
{
lcox.addActionListener(new ActionListener() {
public void actionPerformed(ActionEvent eint)
{
clex.addActionListener(new ActionListener() {
public void actionPerformed(ActionEvent eint)

```



```

{
clex_actionPerformed(eint);
}
});
nimx.addActionListener(new ActionListener() {
public void actionPerformed(ActionEvent eint)
{
try
{
fint.connection(uinx,neow);
}
catch (Exception einw)
{
einw.printStackTrace();
}
}
});
lint.setLayout(null);
jlem.setLayout(null);
addComponent(lint,jlem , 9,30,600,300);
addComponent(jlem, ulem, 90,30,100,19);//1
addComponent(jlem, clem, 90,63,100,19);
addComponent(jlem, lety, 190,30,100,19);
addComponent(jlem, lcom, 190,63,100,19);
addComponent(jlem, lotm, 300,30,90,19);//1
// addComponent(jlem, clef, 111,30,100,28);//1
addComponent(jlem, clef, 300,63,90,19);
addComponent(jlem, cent, 300,93,90,19);
//jlem.setBackground(new Color(174,237, 154));

```

```

jlem.setBackground(new Color(255,255,255));
// lint.setBackground(new Color(30,60,90));

addComponent(jlem, ulex, 90,30,100,19);
// addComponent(jlem, intl, 100,105,70,18);
addComponent(jlem, cldm, 90,63,100,19);
//addComponent(jlem, clon, 100,150,70,18);
addComponent(jlem, letw, 190,30,100,19);//
//addComponent(jlem, inte, 200,105,100,25);
//addComponent(jlem, fldm, 200,150,100,25);
addComponent(jlem, letm, 190,63,100,19);
addComponent(jlem, nlex, 90,123,100,19);
addComponent(jlem, neoy, 190,123,100,19);
//
addComponent(jlem, maot, 90,93,100,19);
addComponent(jlem, asount, 190,93,100,19);
addComponent(jlem, neck, 390,220,100,19);//name
addComponent(jlem,clex, 300,63,100,19);
addComponent(jlem,nimx,300,123,100,28);
addComponent(jlem,lcox,300,30,100,19);
//          addComponent(jlem,neck,300,150,100,28);

neoy.addActionListener(new ActionListener() {
public void actionPerformed(ActionEvent eint)
{
node_actionPerformed(eint);
}
});
this.setTitle("Login");

```

```

this.setSize(new Dimension(600,390));
this.setDefaultCloseOperation(DO_NOTHING_ON_CLOSE);
this.setResizable(xint);
}

/** Add Component Without a Layout Manager (Absolute Positioning) */
private void addComponent(Container mcot,Component cint,int intx,int inty,int
widx,int heim)
{
cint.setBounds(intx,inty,widx,heim);
mcot.add(cint);
}
private void logm_actionPerformed(ActionEvent eint)
{
try
{
uinx=lety.getText();
ncot=lcom.getText();
intf=inte.getText();
cost=asount.getText();//change
if(!uinx.equals("") || !ncot.equals(""))
{
fint.valid(uinx,ncot,intf,cost);
dispose();
}
else
fint.ulongin(uinx,newx,inof,lemt,xval,yval);
}
else

```

```

{
OptionPane.showMessageDialog(null,"give all details ");
}
}
catch (Exception einx)
{
}
System.out.println("\nlogin_actionPerformed(ActionEvent eint) called.");

```

Verification

```

import java.io.*;
import javax.swing.*;
import java.awt.event.*;
import org.gui.JDirectoryDialog; // External Tool for Jfile Chooser
class show extends JFrame implements ActionListener
{
JButton jcut;
JButton jcux;
JButton jcuy;
static String eint="",fint="";
static int mint,xint;
dec necx=new dec();
JPanel nect=new JPanel();
static JTextArea texm=new JTextArea();
JScrollPane nimx=new JScrollPane(texm);
JLabel yint=new JLabel();
static byte ncot[];
static String kint="",dint="",wint="",ecny="";
String dlc="";

```

```

File timx;
public JDirectoryDialog dict;
public show(int finx,byte ncox[],String uint,String finw,String ncme,String
ecoy)throws Exception
{
jcut=new JButton();
jcux=new JButton();
jcuy=new JButton();
jcut.setText("Save");
jcux.setText("Cancel");
jcuy.setText("Dec");
nect.setLayout(null);
ncot=ncox;
kint=ncme;
mint=finx;
fint=finw;
dint=uint;
xint=finx;
ecny=ecoy;
if(finx==1)
{
nect.add(nimx);
nect.add(jcut);
nect.add(jcux)
nimx.setBounds(5,5,390,350);
jcut.setBounds(75,400,100,25);
jcux.setBounds(200,400,100,25);
jcuy.setBounds(75,400,100,25);
texm.setText(necx.Dec(new String(ncot)));

```

```

    }
    else
    {
        nect.add(yint);
        nect.add(jcut);
        nect.add(jcux);
        nect.add(jcuy);

        yint.setBounds(5,5,390,25);
        jcut.setBounds(75,400,100,25);
        jcux.setBounds(200,400,100,25);
        jcuy.setBounds(75,400,100,25);
        yint.setText(necx.Dec(new String(ncot)));
    }
    add(nect);
    jcut.addActionListener(this);
    jcux.addActionListener(this);
    jcuy.addActionListener(this);
    jcut.setVisible(false);
    setSize(500,500);
    setVisible(true);
    setTitle("Node : "+uint+"Received");
    setDefaultCloseOperation(DO_NOTHING_ON_CLOSE);
}

public void actionPerformed(ActionEvent einy)
{
    try
    {
        if(einy.getSource()==jcut)

```

```

{
if(dict == null)
{
dict = new JDirectoryDialog(show.this);
}
if(dict.showDirectoryDialog())
{
File dect = dict.getSelectedFolder();
dlcn=dect.getAbsolutePath();
}
timx=new File(dlcx,kint);
if(timx.exists())    JOptionPane.showMessageDialog(null,"File Already
Exist");
else
{
FileOutputStream finy=new FileOutputStream(timx,true);
finy.write(ncot);
finy.close();
}
dispose();
}
}

```

Set

```

import java.sql.*;
import java.io.*;
import java.net.*;
import java.util.*;
public class set

```

```

{
public dat dint;
public String wint;
ResultSet lint;
Connection cint;
Statement timx,mint;
Vector vint,ncot;
static ServerSocket ncoy;
static Socket clim,netx;
ObjectInputStream dinx,dicx;
ObjectOutputStream dcnx,dcny,ocnt;
InputStream nicx;
OutputStream niex;
link fine;
String
nimy="",metx="",femt,ncox,ncex[],ncow,leto="127.0.0.1",fcox[],mcox,ucox,xc
ow,jcox[],ucow;
String icow[],icom="";
static set sect;
int inet=9000,ieint,inex=1,jaso=0;
public set()throws Exception
{
super();
dint = new dat();
timx=dint.dbc();
mint=dint.dbc();
wint="update undetil set fclient='disable'";
dint.delt(wint);
wint="update jcointm set clientx='disable'";

```



```

dint.delt(wint);
wint="update undetil set mclient='0'";
dint.delt(wint);
wint="update undetil set ycol='Node'";
dint.delt(wint);
wint="delete from flowetn";
dint.delt(wint);
}

public void listen()throws Exception
{
    clim=ncoy.accept();
    dinx=new ObjectInputStream(clim.getInputStream());
    String femt=(String)dinx.readObject();
    System.out.println("listen:"+femt);
    if(femt.equals("uinx"))
    {
        String oint="";
        dinx=new ObjectInputStream(clim.getInputStream());
        String ncem=(String)dinx.readObject();
        String nimy[]=ncem.split("&");
        int jnet=Integer.parseInt(nimy[1]);
        wint="select * from undetil where xclient='"+nimy[0]+'";
        if(dint.nceo(wint))
        {
            oint="exist";
            wint="delete from undetil where xclient="";

            dint.delt(wint);
        }
    }
}

```

```

else
{
wint="update undetil set  xclient='"+nimy[0]+" where mclient='"+jnet+""";
dint.inst(wint);
wint="update undetil set  wclient='"+nimy[3]+" where mclient='"+jnet+""";
dint.inst(wint);
wint="update undetil set  dclient='"+nimy[4]+" where mclient='"+jnet+""";
dint.inst(wint);

wint="update undetil set  oclient='10' where mclient='"+jnet+""";
dint.inst(wint);
wint="update undetil set  xcol=" where mclient='"+jnet+""";
dint.inst(wint);
wint="update undetil set  ycol='Node' where mclient='"+jnet+""";
dint.inst(wint);
wint="update undetil set  mkey='"+nimy[9]+" where mclient='"+jnet+""";
dint.inst(wint);
wint="update undetil set  skey='"+nimy[10]+" where mclient='"+jnet+""";
dint.inst(wint);
wint="update undetil set  ekey='"+nimy[14]+" where mclient='"+jnet+""";
dint.inst(wint);
oint="ok";
}
cnx=new ObjectOutputStream(clim.getOutputStream());
dcnx.writeObject(oint);
}
Vector fint=new Vector();
String ncej=(String)dinx.readObject();
System.out.println("femt detm ncej1: "+ncej);

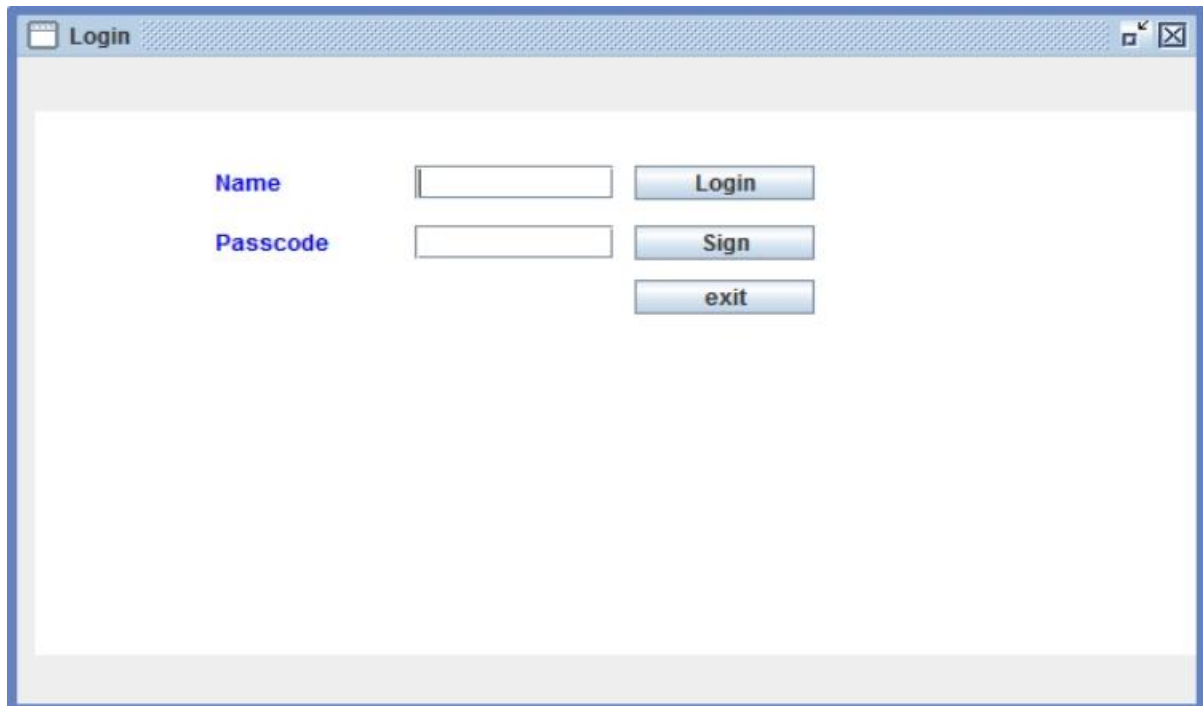
```

```

if(!ncej.equals(""))
{
wint="select xclient from undetil where xclient!=""+ncej+" and
fclient='enable'";
}
else
{
wint="select xclient from undetil where xclient!=""+ncej+""";
System.out.println("femt detm ncej2: "+ncej);
}
lint=timx.executeQuery(wint);
System.out.println("femt detm ncej3: "+ncej);
System.out.println("femt detm ncej3 lint: "+lint);
fint.add("select");
while(lint.next())
{
String ncme=lint.getString(1);
System.out.println("ncej3 ncme: "+ncme);
if(!ncme.equals(""))
fint.add(ncme);
}
wint="update undetil set mkey='"+nimy[9]+" where mclient='"+jnet+""';
dint.inst(wint);
wint="update undetil set skey='"+nimy[10]+" where mclient='"+jnet+""';
dint.inst(wint);
dcnx=new ObjectOutputStream(clim.getOutputStream());
dcnx.writeObject(fint);
}

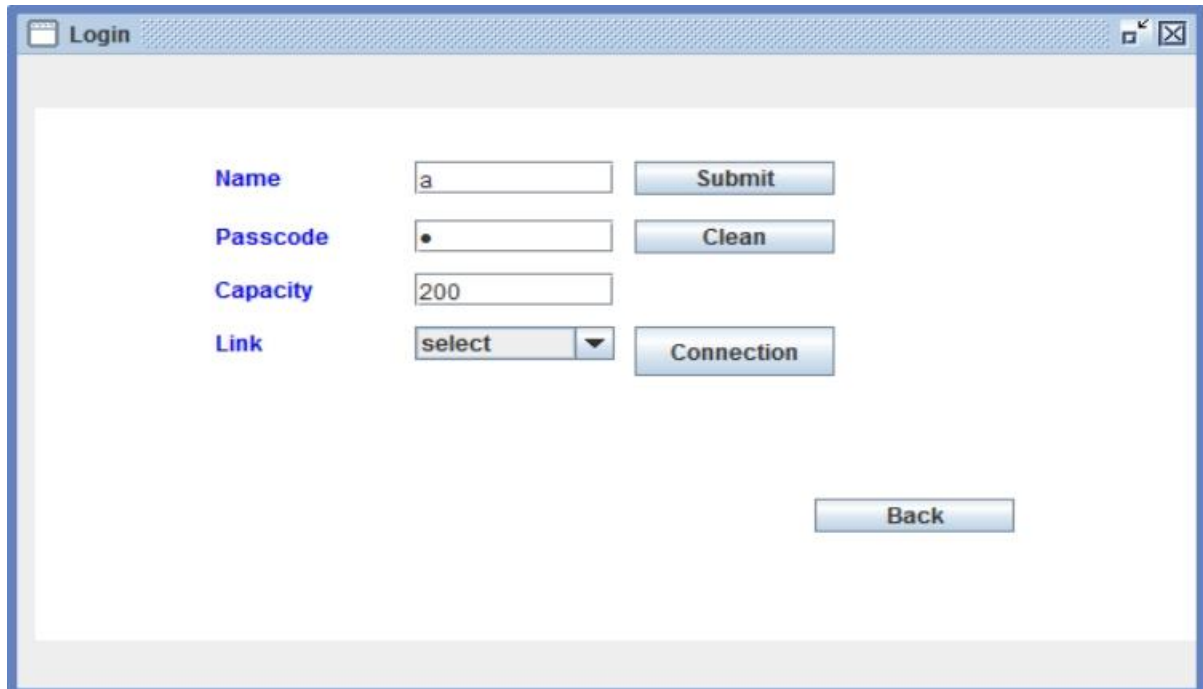
```

7.2 SAMPLE SCREEN SHOTS



The screenshot shows a window titled "Login". Inside the window, there are two labels: "Name" and "Passcode". Next to "Name" is a text input field. Next to "Passcode" is a text input field. To the right of the "Name" input field is a button labeled "Login". To the right of the "Passcode" input field are two buttons: "Sign" and "exit".

Fig 7.1 Node Login



The screenshot shows a window titled "Login". Inside the window, there are four labels: "Name", "Passcode", "Capacity", and "Link". Next to "Name" is a text input field containing the letter "a". Next to "Passcode" is a text input field containing a single dot. Next to "Capacity" is a text input field containing the number "200". Next to "Link" is a dropdown menu with the word "select" and a downward arrow. To the right of the "Name" input field is a button labeled "Submit". To the right of the "Passcode" input field is a button labeled "Clean". To the right of the "Link" dropdown menu is a button labeled "Connection". At the bottom right of the window is a button labeled "Back".

Fig 7.2 New Node Creation



Fig 7.3(a) Master Key Generation

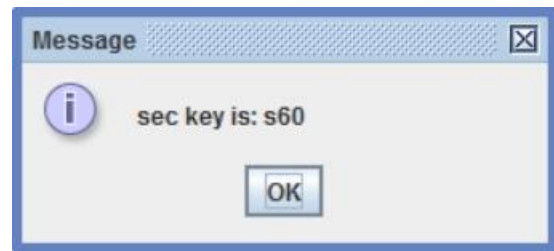


Fig 7.3(b) Secret Key Generation



Fig 7.3(c) Encryption Key Generation

Fig 7.3 Key Generation

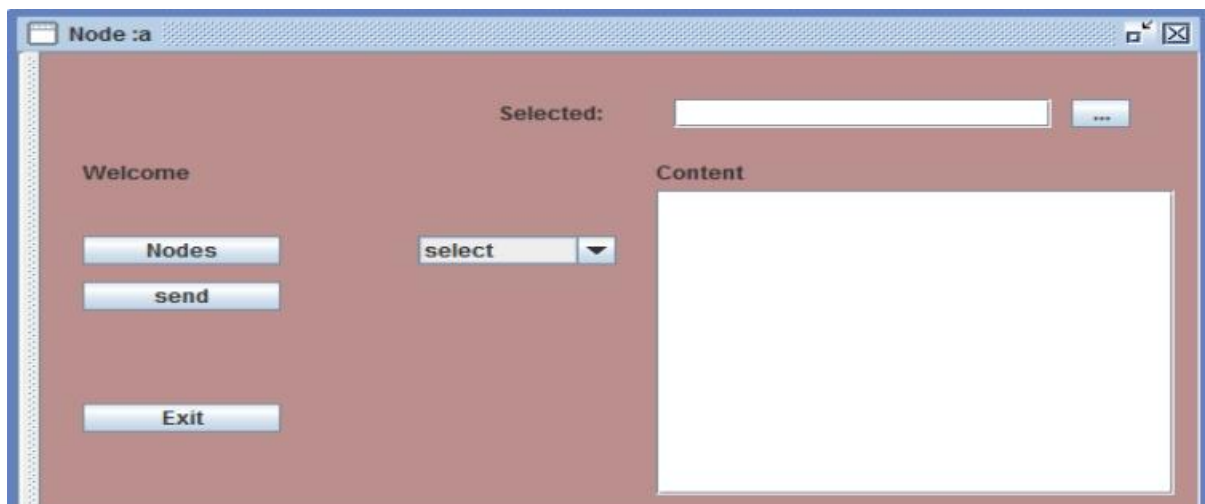
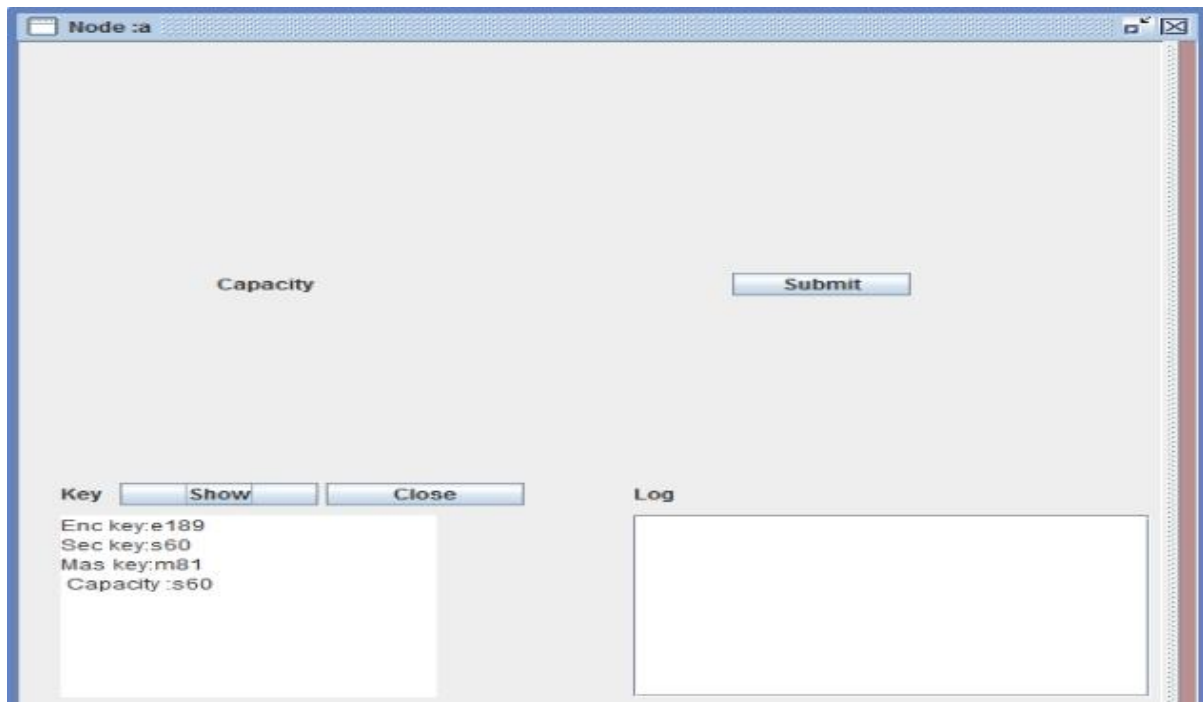
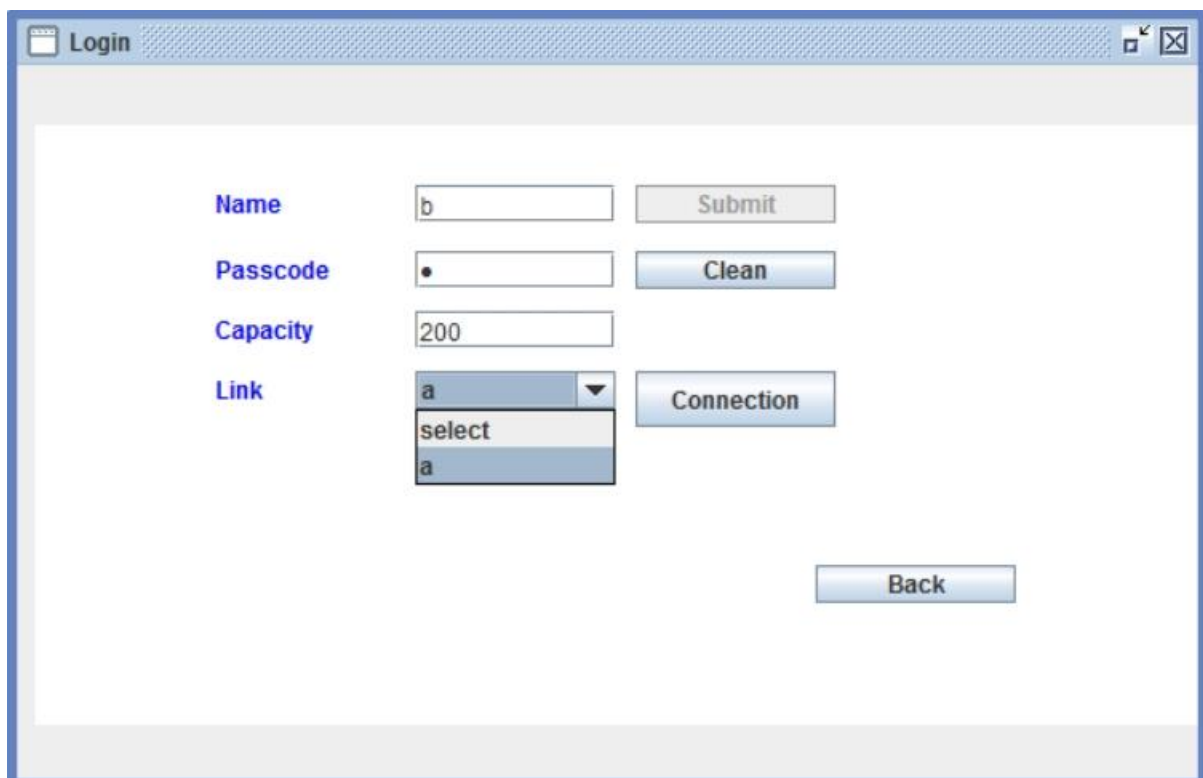


Fig 7.4 Node Login Success



The 'Node :a' window contains a 'Capacity' label and a 'Submit' button. At the bottom left, there is a 'Key' section with 'Show' and 'Close' buttons, and a text area displaying: 'Enc key:e189', 'Sec key:s60', 'Mas key:m81', and 'Capacity :s60'. To the right of the 'Key' section is a 'Log' text area.

Fig 7.5 Node Configuration



The 'Login' window features four input fields on the left: 'Name' (containing 'b'), 'Passcode' (containing a dot), 'Capacity' (containing '200'), and 'Link' (a dropdown menu with 'a' selected and a list showing 'select' and 'a'). To the right of these fields are four buttons: 'Submit' (next to Name), 'Clean' (next to Passcode), 'Connection' (next to Link), and 'Back' at the bottom right.

Fig 7.6 Node Connection

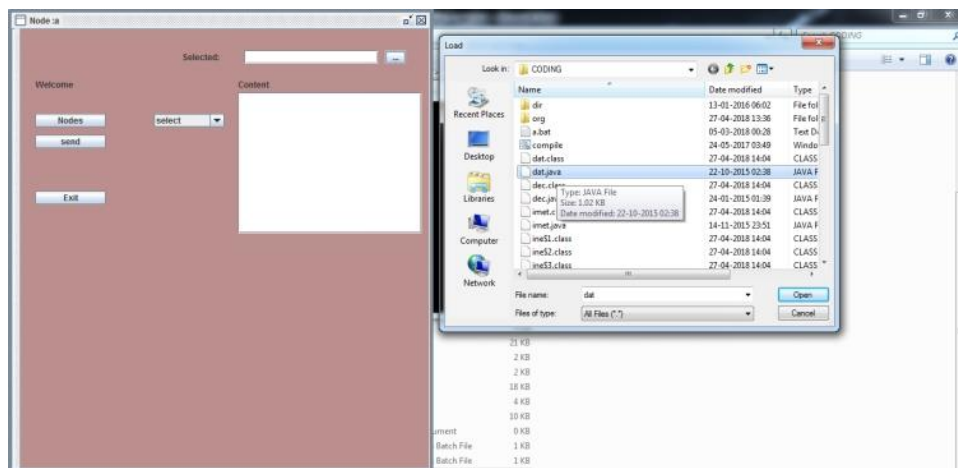


Fig 7.7 File Selection



Fig 7.8 Encryption by Source Node

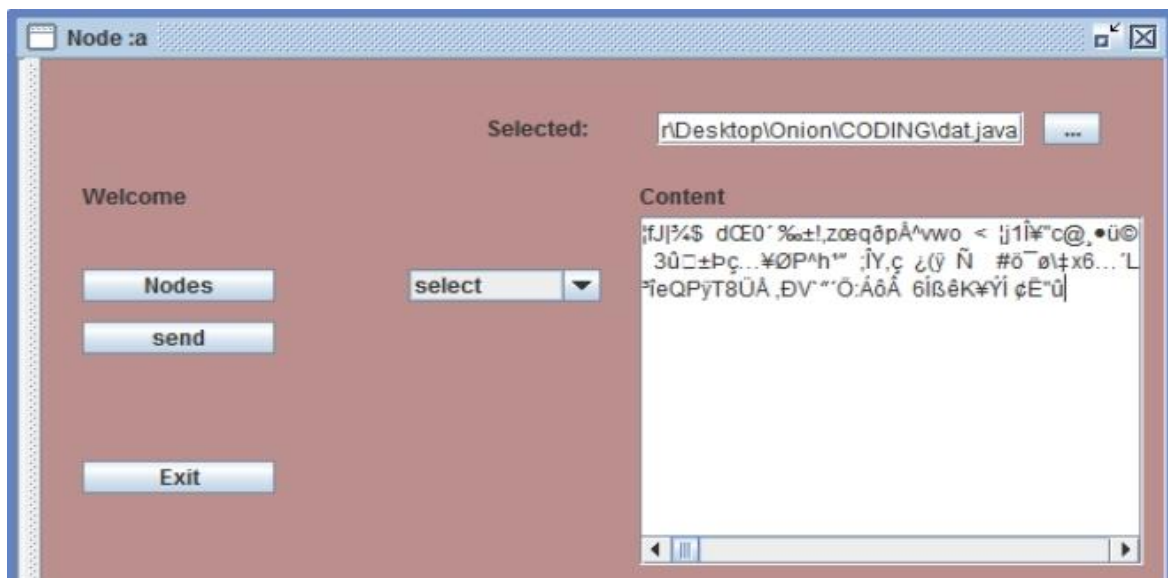


Fig 7.9 File After Encryption

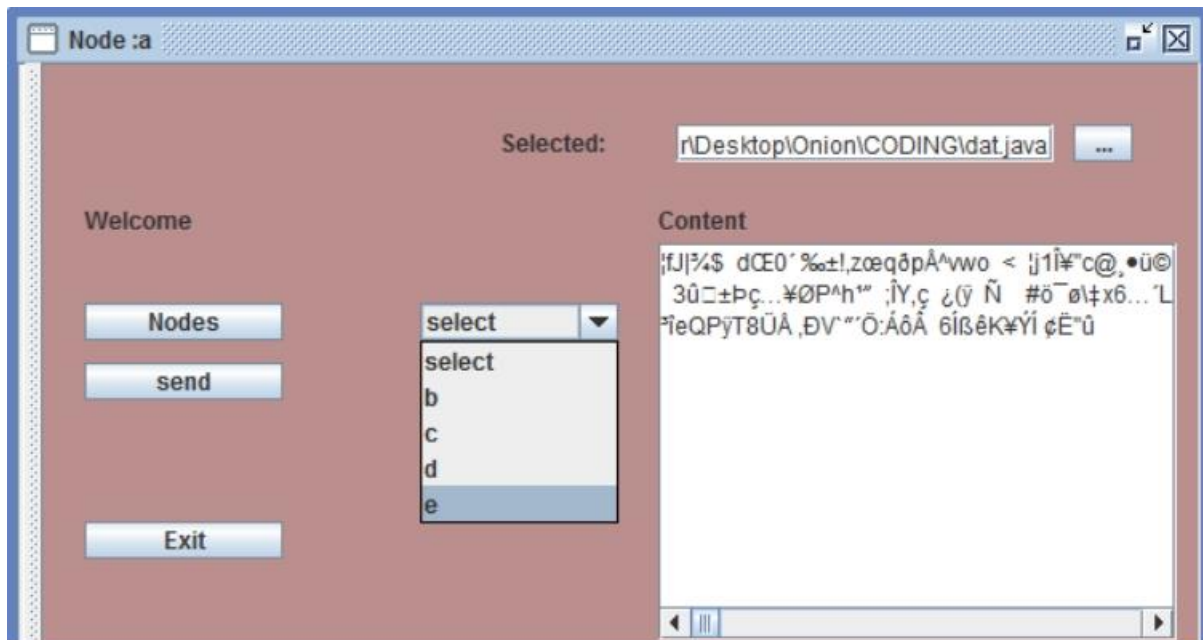


Fig 7.10 Selection of Destination Node

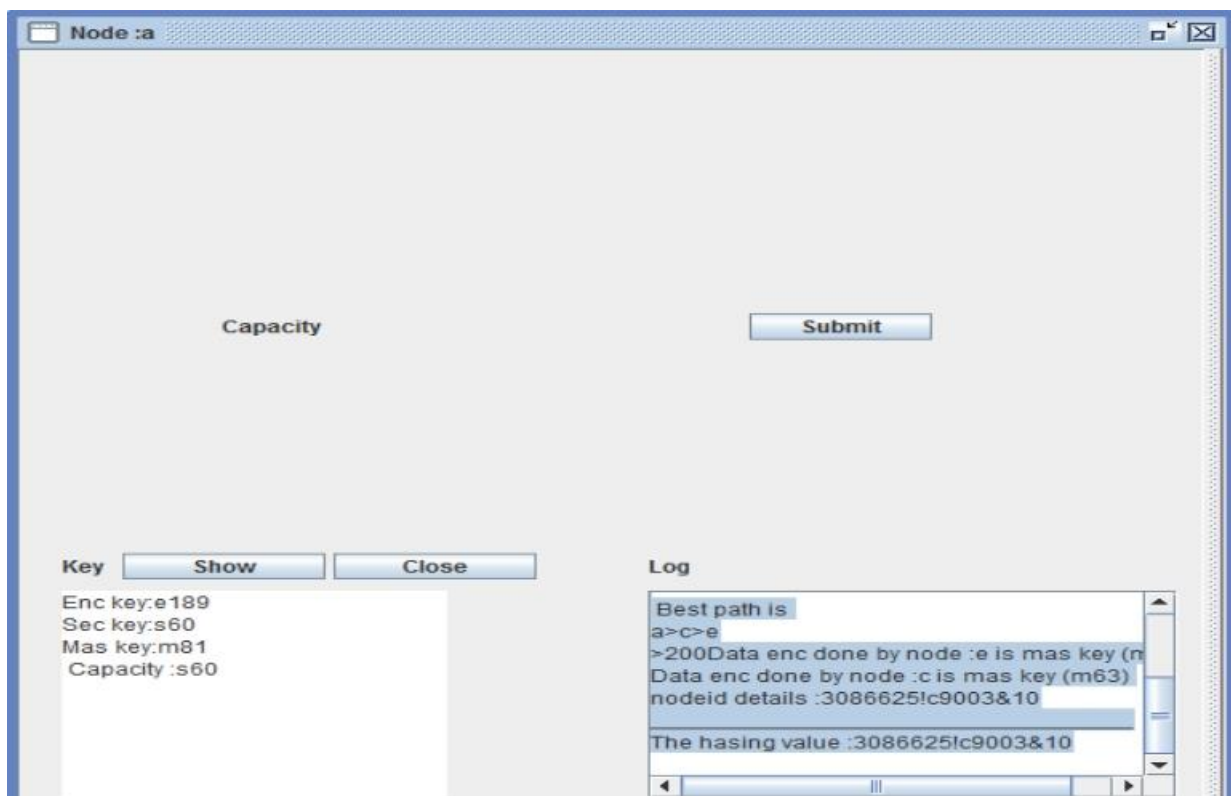


Fig 7.11 Best Path Selection

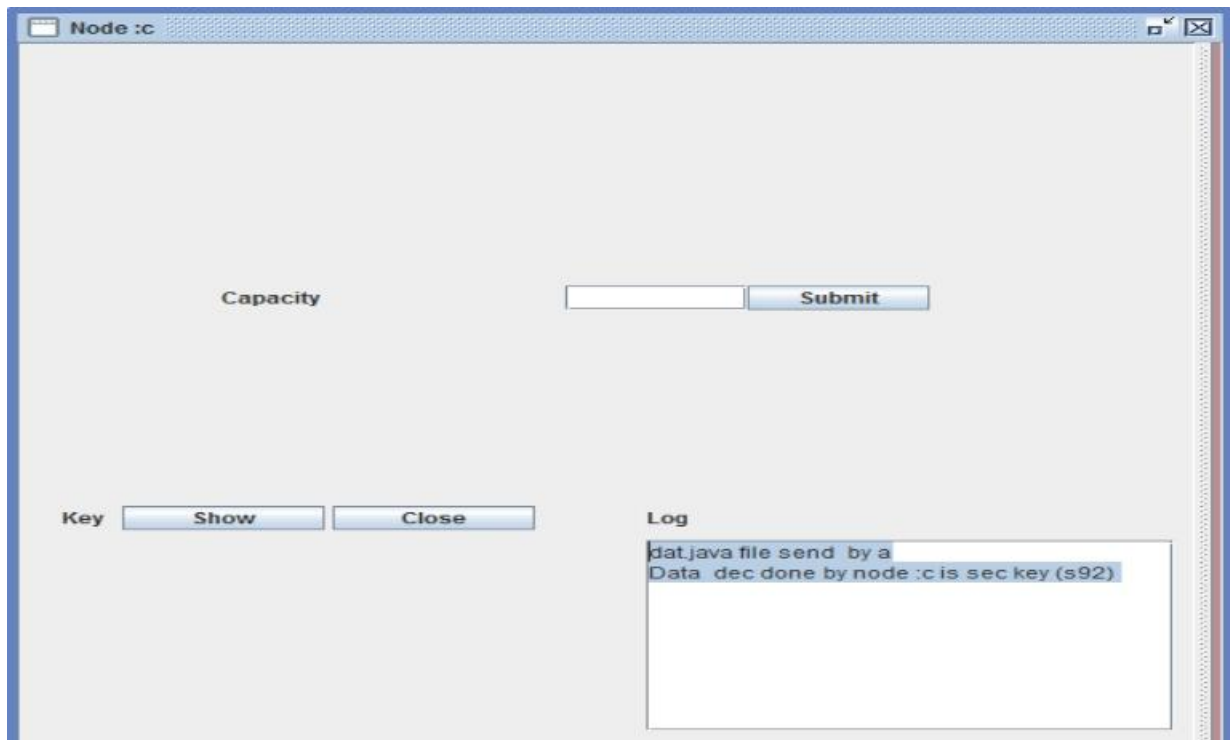


Fig 7.12 Decryption by Intermediate Node

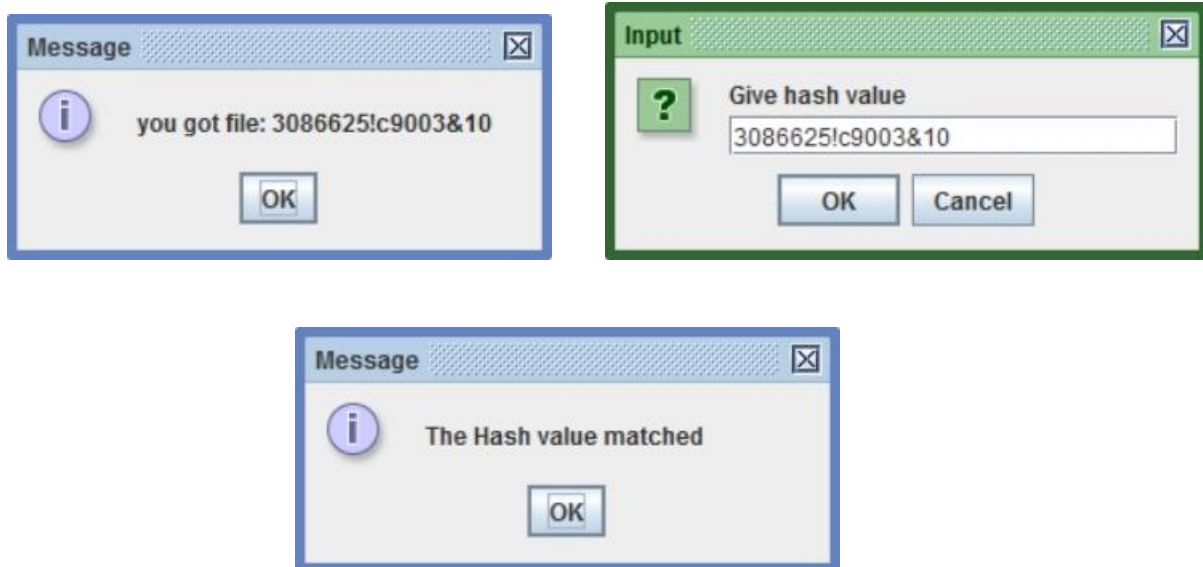


Fig 7.13 Intermediate Node ID Verification

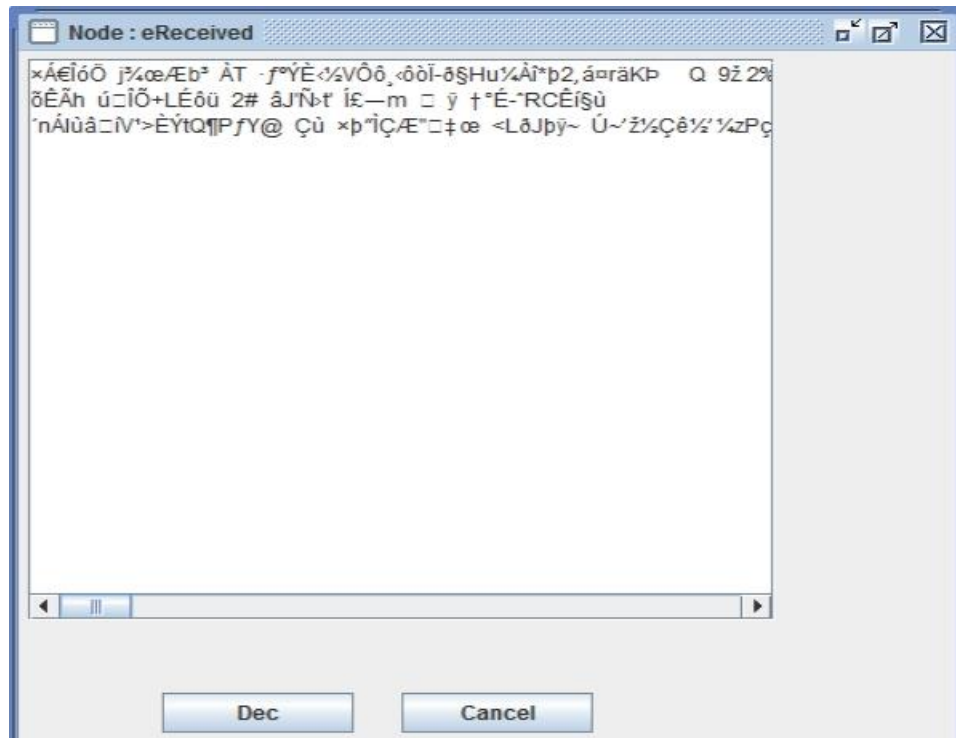


Fig 7.14 Encrypted Data Received at Destination Node

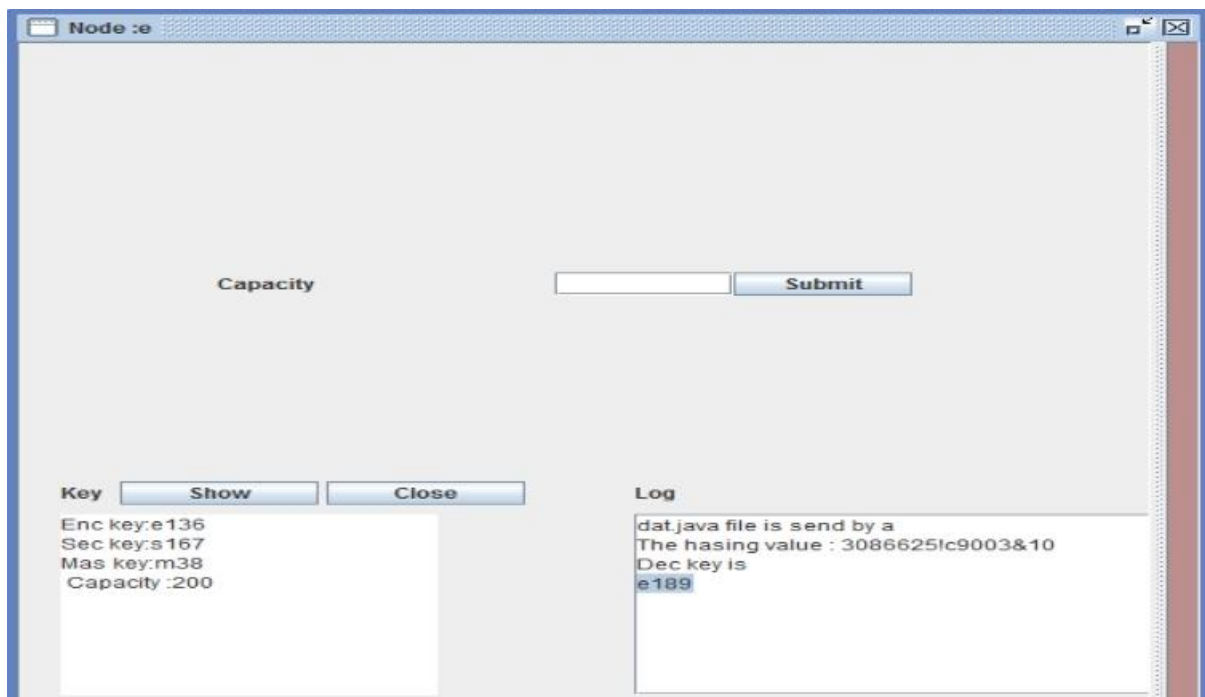


Fig 7.15 Destination Decryption Key Verification

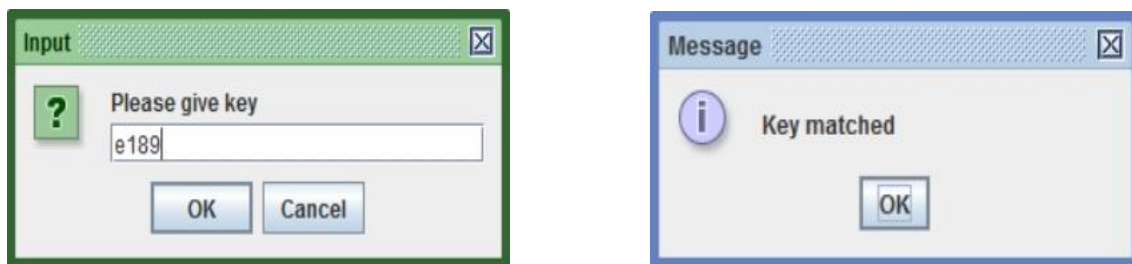


Fig 7.16 Decryption Key Verification

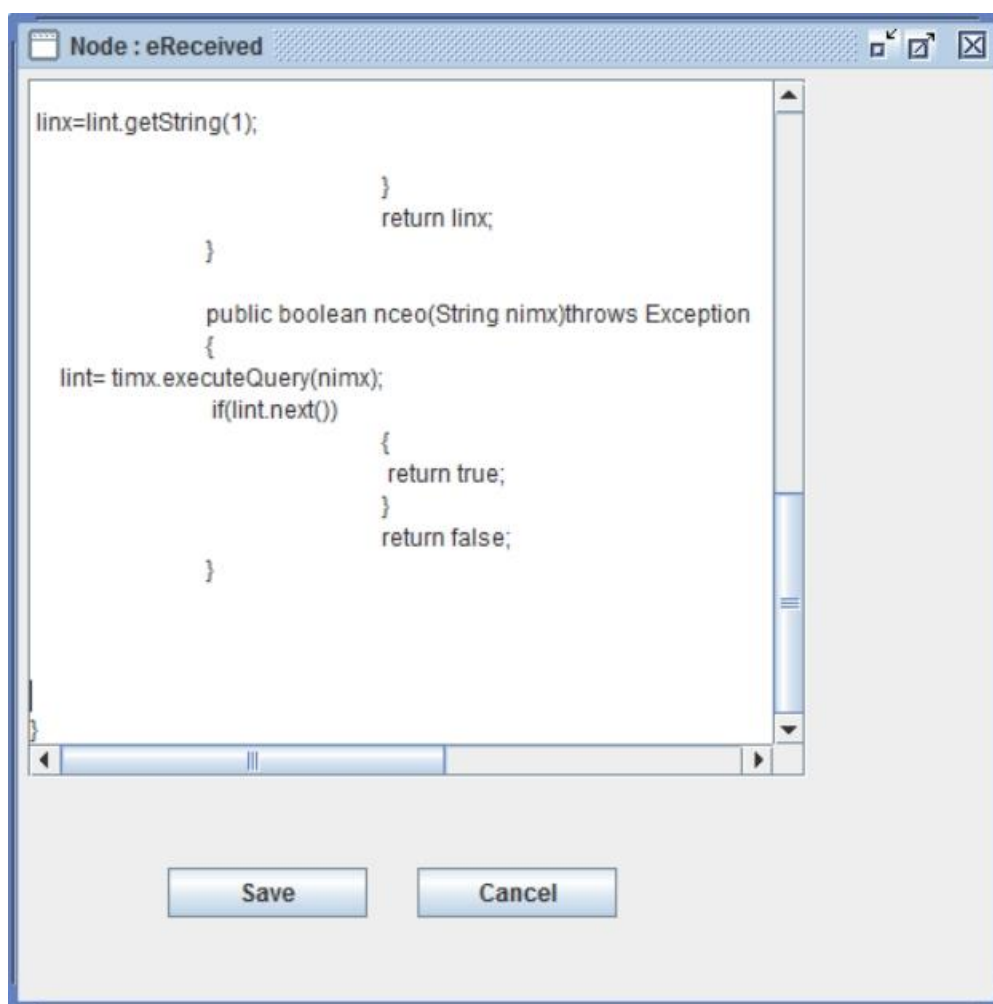


Fig 7.17 Original Data at Destination Node

TESTING AND MAINTANENCE

Input Design

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system.

The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy.

Input Design considered the following things

- What data should be given as input?
- How the data should be arranged or coded?
- The dialog to guide the operating personnel in providing input.
- Methods for preparing input validations and steps to follow when error occur.

Output Design

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

The output form of an information system should accomplish one or more of the following objectives.

- Convey information about past activities, current status or projections of the future.
- Signal important events, opportunities, problems, or warnings.
- Trigger an action.
- Confirm an action.

Test plan

Software testing is the process of evaluation a software item to detect differences between given input and expected output. Also to assess the feature of a software item. Testing assesses the quality of the product. Software testing is a process that should be done during the development process. In other words software testing is a verification and validation process.

Verification

Verification is the process to make sure the product satisfies the conditions imposed at the start of the development phase. In other words, to make sure the product behaves the way we want it to.

Validation

Validation is the process to make sure the product satisfies the specified requirements at the end of the development phase. In other words, to make sure the product is built as per customer requirements.

8.1 BLACK BOX TESTING

Black box testing is a testing technique that ignores the internal mechanism of the system and focuses on the output generated against any input and execution of the system. It is also called functional testing.

8.2 WHITE BOX TESTING

White box testing is a testing technique that takes into account the internal mechanism of a system. It is also called structural testing and glass box

testing. Black box testing is often used for validation and white box testing is often used for verification.

8.3 UNIT TESTING

Unit testing is the testing of an individual unit or group of related units. It falls under the class of white box testing. It is often done by the programmer to test that the unit he/she has implemented is producing expected output against given input.

8.4 INTEGRATION TESTING

Integration testing is testing in which a group of components are combined to produce output. Also, the interaction between software and hardware is tested in integration testing if software and hardware components have any relation. It may fall under both white box testing and black box testing.

Functional Testing

Functional testing is the testing to ensure that the specified functionality required in the system requirements works. It falls under black box testing.

8.5 SYSTEM TESTING

System testing is the testing to ensure that by putting the software in different environments (e.g., Operating Systems) it still works. System testing is done with full system implementation and environment. It falls under the class of black box testing.

Stress Testing

Stress testing is the testing to evaluate how system behaves under unfavorable conditions. Testing is conducted at beyond limits of the specifications. It falls under the class of black box testing.

Performance Testing

Performance testing is the testing to assess the speed and effectiveness of the system and to make sure it is generating results within a specified time as in performance requirements. It falls under the class of black box testing.

Usability Testing

Usability testing is performed to the perspective of the client, to evaluate how the GUI is user-friendly? How easily can the client learn? After learning how to use, how proficiently can the client perform? How pleasing is it to use its design? This falls under the class of black box testing.

8.6 ACCEPTANCE TESTING

Acceptance testing is often done by the customer to ensure that the delivered product meets the requirements and works as the customer expected. It falls under the class of black box testing. It is a critical part of a project report that involves evaluating whether the project's deliverables meet the customer's requirements and expectations. The purpose of acceptance testing is to ensure that the project meets the agreed-upon specifications, functions as intended and is ready for deployment.

Regression Testing

Regression testing is the testing after modification of a system, component, or a group of related units to ensure that the modification is working correctly and is not damaging or imposing other modules to produce unexpected results. It falls under the class of black box testing.

The goal of regression testing is to catch any defects or errors that may have been introduced as a result of changes to the system, and to ensure that the system continues to perform as expected. This type of testing is particularly important when changes are made to a large and complex system, as it can help to identify any unintended impacts of those changes on other parts of the system.

8.7 TEST CASE SPECIFICATION

Table 8.1 Test Case Specification

| TEST CASE ID | TEST CASE TITLE | STEPS | EXPECTED RESULT | ACTUAL RESULT | STATUS |
|---------------------|--------------------------------|--|--|--|---------------|
| TC1 | Node Registration | 1. Enter the name, capacity, link, password of node. 2. Click signin. | Node successfully created. | Node successfully created. | Pass |
| TC2 | Node Login | 1. Enter the node details. 2. Click Login. | Login Success. | Login Success. | Pass |
| TC3 | Multilayer Encryption | 1. select the file and destination node. 2. Click on send. | Data is encrypted at multiple level before reaching destination. | Data is encrypted at multiple level before reaching destination. | Pass |
| TC4 | Intermediate node verification | 1. Hash each of the intermediate nodes's id in the network and verify. | Hash value matched. | Hash value matched. | Pass |
| TC5 | Decryption | 1. After receiving data click on decrypt 2. Enter decryption key | Data decrypted successfully | Data decrypted successfully | Pass |

CONCLUSION AND FUTURE ENHANCEMENTS

9.1 CONCLUSION

In the Proposed Work, we can have implemented Multi Layer Security system for effective data Transfer from Source to destination through Multi Hop connectivity. We have a Complex Multilayer secured Protocol for highly secured data transfer. We also implemented Encryption of data using RSA apart from Multi Layer Encryption. In conclusion, the multilayer encryption project is a complex system that provides high-level security for sensitive data. The project involves using multiple layers of encryption algorithms, such as RSA and SHA-256, to ensure that the data is protected from unauthorized access and hacking attempts. The use of multiple encryption layers provides an additional layer of protection against attacks and helps to ensure that the data remains secure even if one layer is compromised.

9.2 FUTURE ENHANCEMENTS

The Future work of this Project is dynamic route Connecting when any Intermediate Node is failed. Apart from Capacity calculation, we can also include Throughput, Energy for choosing Best Route / Hop for effective data Transfer. As technology evolves, new encryption algorithms may become available that are stronger or more efficient than the ones currently being used. By incorporating these new algorithms into the multilayer encryption project, the overall security of the system can be improved. Homomorphic encryption is a relatively new technique that allows data to be encrypted and processed without being decrypted. This can be useful in scenarios where sensitive data needs to be processed by third-party applications or services without exposing the data to those services.

REFERENCES

1. Al-Saraireh, A. A. Al-Ahmad, and A. M. Al-Fayoumi. (2010). "Multilayer encryption: An approach to enhance network security". *International Journal of Computer Science and Security*, 4(2), 203-212.
2. Al-Fayoumi, N. H., Al-Saraireh, A. J., & Al-Ahmad, A. A. (2012). "Multilayer Encryption and Steganography: An Effective Data Hiding Technique". *Journal of Computer Science*, 8(1), 17-25.
3. Chou, D. Wei, C. Kuo, and K. Naik, "An Efficient Anonymous Communication Protocol for Peer-to-Peer Applications over Mobile Ad-Hoc Networks," *IEEE J. Selected Areas in Comm.*, vol. 25, no. 1, Jan. 2007.
4. Diaz, C., & Seys, S. (2015). "The TOR Network: Anonymity and Privacy on the Internet". In *Privacy Enhancing Technologies* (pp. 241-259). Springer, Cham.
5. Dingledine, R., Mathewson, N., & Syverson, P. (2004). "Tor: The Second-Generation Onion Router". In *Proceedings of the 13th USENIX Security Symposium* (pp. 303-320). USENIX Association.
6. G. Indirania and K. Selvakumara, "A Swarm-Based Efficient Distributed Intrusion Detection System for Mobile Ad Hoc Networks (MANET)," *Int'l J. Parallel, Emergent and Distributed Systems*, vol. 29, pp. 90-103, 2014.
7. G. Shen, J. Liu, D. Wang, J. Wang, and S. Jin, "Multi-Hop Relay for Next-Generation Wireless Access Networks," *Bell Labs Technical J.*, vol. 13, no. 4, pp. 175-193, 2009.

8. H. Li and M. Singhal, "Trust Management in Distributed Systems," *Computer*, vol. 40, no. 2, pp. 45-53, Feb. 2007.
9. J. Appelbaum, R. Dingledine, and N. Mathewson, "Tor: Design and Implementation of a Tor-based Anonymous Communication System," in *Proceedings of the 13th ACM Conference on Computer and Communications Security (CCS '06)*, Alexandria, VA, USA, 2006, pp. 86-93, doi: 10.1145/1298455.1298478.
10. J. Jaiswal, M. K. Soni, and N. K. Shukla, "Multilayer encryption using Advanced Encryption Standard (AES) and Rivest-Shamir-Adleman (RSA) algorithm," *International Journal of Computer Science and Information Security*, vol. 11, no. 2, pp. 27-32, 2013.
11. Johnson, A., Syverson, P., & Dingledine, R. (2006). "Onion routing for anonymous and private internet connections". *Communications of the ACM*, 52(2), 36-41. doi: 10.1145/1113034.1113062.
12. K. Liu, J. Deng, and K. Balakrishnan, "An AcknowledgementBased Approach for the Detection of Routing Misbehavior in MANETs," *IEEE Trans. Mobile Computing*, vol. 6, no. 5, pp. 536- 550, May 2007.
13. M. Mahmoud and X. Shen, "ESIP: Secure Incentive Protocol with Limited Use of Public-Key Cryptography for Multi-Hop Wireless Networks," *IEEE Trans. Mobile Computing*, vol. 10, no. 7, pp. 997-1010, July 2011.
14. M. Mahmoud and X. Shen, "PIS: A Practical Incentive System For Multi-Hop Wireless Networks," *IEEE Trans. Vehicular Technology*, vol. 59, no. 8, pp. 4012-4025, Oct. 2010.
15. McCoy, D., Levchenko, K., & Voelker, G. M. (2012). "The dark side of the onion: Examining the harmful side effects of Tor's hidden services". In *Proceedings of the 2012 ACM conference on Computer and communications security* (pp. 277-288).