VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"Jnana Sangama", Belagavi – 560 018.

A FINAL PROJECT REPORT

ON

"DESIGN AND ANALYSIS OF CRYPTOGRAPHY COMMUNICATION SYSTEM"

Submitted in partial fulfillment for the award of

Bachelor of Engineering in Computer Science and Engineering

By

CHANDU BHARGAVI K R	**1CK22CS035**
DEEPIKA U	**1CK22CS040**
GANASHREE J	**1CK22CS046**
GAYANA G	**1CK22CS048**

Under the Guidance of

Dr. Ramesh Kumar V
Professor & HOD,
Dept. of CE, CBIT.

C. BYREGOWDA INSTITUTE OF TECHNOLOGY

Approved by AICTE New Delhi, Accredited by NAAC with B++ Grade, Recognized by Govt. of Karnataka, Affiliated to VTU Belagavi, ISO 9001:2015 Certified Institute

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Kolar – Srinivaspur Road, Kolar – 563101
2025-2026

C BYREGOWDA INSTITUTE OF TECHNOLOGY

Approved by AICTE New Delhi,Accredited by NAAC with B++ Grade, Recognized by Govt. of Karnataka, Affiliated to VTUBelagavi, ISO 9001:2015 Certified Institute

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Kolar-Srinivasapura road, Kolar-563101

CERTIFICATE

This is to certify that the Final project work entitled "**DESIGN AND ANALYSIS OF CRYPTOGRAPHY COMMUNICATION SYSTEM**" is a bonafied work carried out by **CHANDU BHARGAVI K R (USN:1CK22CS035), DEEPIKA U (USN:1CK22CS040), GANASHREE J (USN:1CK22CS046),** and **GAYANA G(USN:1CK22CS048)** in partial fulfillment for the award of Bachelor of Engineering in Computer Science & Engineering of the Visvesvaraya Technological University, Belagavi during the year 2025-26. It is certified that all corrections/suggestions indicated for the internal assessment have been incorporated in the report. The project report has been approved as it satisfies the academic requirements in respect of project work prescribed for the VII Semester Bachelor of Engineering Degree.

Signature of the Guide	Signature of the HOD	Signature of the Principal
Dr. Ramesh Kumar V	**Dr. Vasudeva R**	**Dr. S N Chandrashekara**
Professor & HOD	**Assoc. prof. & HOD,**	**Principal, CBIT**
Dept. of CE, CBIT	**Dept. of CSE, CBIT**	

External Viva

Examiner Name	**Signature with Date**
1. …………………………	**…………………………**
2. …………………………	**…………………………**

ABSTRACT

The rapid advancement of digital communication technologies has increased the need for secure and privacy-preserving messaging systems. Conventional chat applications primarily focus on speed and convenience, often relying on static encryption techniques that do not adapt to user context or evolving security threats. This project addresses these limitations by proposing a Secure AI-Based Chat System that enhances communication security through intelligent and adaptive mechanisms.

The proposed system integrates artificial intelligence to analyze the emotional content of user messages and applies emotion-aware encryption to protect data during transmission. Along with secure text messaging, the system supports encrypted file sharing, hidden data transmission using steganography, and secure storage of chat logs. A decoy communication mode is also implemented to safeguard users in sensitive situations by displaying misleading messages while securely preserving actual conversations.

The application is developed using Python with a graphical user interface that ensures ease of use and cross-platform compatibility. Testing results show that the system provides reliable performance, improved security, and user-friendly interaction. By combining AI, encryption, and covert communication techniques, the project demonstrates an innovative and practical approach to building next-generation secure communication systems suitable for academic and real-world applications.

DECLARATION

We, **CHANDUBHARGAVI K R** bearing USN **1CK22CS035**, **DEEPIKA U** bearing USN **1CK22CS040, GANASHREE J** bearing USN **1CK22SCS046, GAYANA G** bearing USN **1CK22CS048,** Students of 7th semester B.E., Computer Science and Engineering of VTU, declare that this project report entitled "**DESIGN AND ANALYSIS OF CRYPTOGRAPHY COMMUNICATION SYSTEM",** embodies the report of the project work carried out under the guidance of **Dr. Ramesh Kumar V, Professor & HOD** Dept., of **CE, CBIT** as partial fulfillment of the requirement of the award of the degree in Bachelor of Engineering, Computer Science and Engineering, affiliated to **Visvesvaraya Technological University, Belagavi** during academic year **2025- 2026.** Further the content embodies in the project has not been submitted previously by anybody for the award of any other degree.

Place: Kolar

Date:

Signature of Student

(**CHANDU BHARGAVI K R)**

(DEEPIKA U)

(GANASHREE J)

(GAYANA G)

ACKNOWLEDGEMENT

The completion of project brings with and sense of satisfaction, but it is never completed without thanking the persons who are all responsible for its successful completion first and foremost.

We wish to express our deep sincere feelings of gratitude to my Institution, **C Byregowda Institute of Technology,** for providing mean opportunity to do our education.

We extend our deep of sincere gratitude to our beloved principal **Dr Chandrashekar SN,** for permitting to carry out the project work successfully**.**

We extend our heartfelt sincere gratitude to our beloved **Associate Professor** and **HOD Dr Vasudeva R,** for his valuable suggestions and support.

We extend our sincere gratitude to our project coordinator **Asst. Professor Kavitha N,** for their valuable advice and constant support.

We extend our sincere gratitude to our internal guide **Dr. Ramesh Kumar V, Professor** and **HOD,**for her suggestions and instructions has served as major contributor towards the project work.

We would like to thank all the Teaching and non-teaching staff members of Department of Computer Science and Engineering, for their support.

Finally, we like to thank our parents and friends for their continuous encouragement and moral support.

CHANDU BHARGAVI K R	1CK22CS035
DEEPIKA U	1CK22CS040
GANASHREE J	1CK22CS046
GAYANA G	1CK22CS048

TABLE OF CONTENTS

Abstract	**I**
Declaration	**II**
Acknowledgement	**III**
Table of Contents	**IV**
List of Figures	**VIII**
List Of Tables	**IX**

CHAPTER	CHAPTER NAME	PAGE NO
1	**Introduction**	**1**
	1.1 Basic topics of the project	2
	1.2 Scope of project	3
	1.3 Motivation	5
	1.4 Objectives	6
	1.5 Organization of report	7
2	**Literature Survey**	**9**
	2.1Introduction	9
	2.2 Related work	10
	2.3 Summary	13
	2.4 Problem Definition	13
3	**System Requirements and Specification**	**14**
	3.1 Functional requirements	15
	3.2 Non-Functional requirements	15
	3.3 Hardware requirements	16
	3.4 Software requirements	16

	3.5 Preliminary Investigation	17
	3.6 System Environment	17
4	**System Analysis**	**18**
	4.3 Existing System	18
	4.4 Proposed System	19
5	**System Design**	**20**
	5.1 Input Design	20
	5.2 Output Design	21
	5.3 System Architecture	22
	5.4 Dataflow Diagram	24
	5.5 Sequence Diagram	25
	5.6 Class Diagram	25
	5.7 Use case Diagram	27
6	**Implementation**	**28**
	6.1 Introduction	28
	6.2 Language selection	28
	6.3 Platform Selection	29
	6.4 Implemented module	29
	6.5 Pseudocode	32
7	**Testing**	**37**
	7.1 Introduction	37
	7.2 Types of Tests	37
	7.3 Test case	38
8	**Results and Discussions**	**42**
	8.1 Discussion	44
9	**Conclusion and Future Enhancement**	**44**
	9.1 Future Enhancement	44

BIBLOGRAPHY 46

APPENDIX-A ACRONYMS 48

APPENDIX-B SNAPSHOTS 49

LIST OF FIGURES

Figure No.	Figure Name	Page No.
4.2	Proposed System	19
5.1	Input Design	21
5.2	Output Design	22
5.3	System Architecture	23
5.4	Dataflow Diagram	24
5.5	Sequence Diagram	25
5.6	Class Diagram	26
5.7	Use case Diagram	27

LIST OF TABLES

Table No	Table Name	Page No
7.1	Unit Test case	39
7.2	Integrated Test Case	39
7.3	System Test Case	40
7.4	Usability Test Case	40
7.5	Security Test Case	41

CHAPTER 1

INTRODUCTION

In the modern digital era, communication technologies have become an integral part of personal, professional, and organizational interactions. With the rapid growth of internet-based messaging platforms, concerns related to data privacy, information leakage, cyber-attacks, and unauthorized access have increased significantly. Conventional chat applications primarily focus on message delivery and user convenience, often neglecting advanced security mechanisms that adapt to the nature of communication. As sensitive information such as confidential documents, intellectual property, and personal conversations are frequently exchanged over digital channels, there is a growing need for intelligent and secure communication systems that provide enhanced protection beyond traditional encryption methods.

This project titled **"**Design and analysis of cryptography communication system**"** aims to address these challenges by integrating artificial intelligence with secure communication techniques. The proposed system introduces an innovative approach where text-based emotion detection is used as a dynamic factor in the encryption process. By analyzing the emotional tone of user messages using AI models such as transformer-based classifiers and sentiment analysis techniques, the system adapts its encryption behavior accordingly. This emotion-driven mechanism adds an additional layer of unpredictability to the encryption process, thereby strengthening message confidentiality and making unauthorized decryption more difficult.

In addition to secure text communication, the system supports secure file transfer and steganography, enabling users to hide confidential text and files within digital images using Least Significant Bit (LSB) techniques. This feature ensures covert data transmission, making the existence of sensitive information less detectable to third parties. The application further enhances security through encrypted chat logs, ensuring that stored communication data remains protected even if local storage is compromised. To counter social engineering threats and forced access scenarios, a Decoy Mode is implemented, which displays fake messages to observers while securely preserving the original conversation in the background.

The proposed Secure AI Chat system is developed using Python, leveraging libraries such as socket programming for network communication**,** CustomTkinter for graphical user interface design**,** machine learning frameworks for emotion analysis**,** and cryptographic techniques for data protection**.**

1.1 Basic Topics of the project

Projects through encrypted communication systems are based on several basic topics that form the backbone of secure data transmission. It starts with an understanding of encryption and includes the purpose, type and application in securing digital communications.

The project validates symmetric and asymmetric encryption technologies such as AES, and messages are devoured and deciphered. This includes:

- **Background of Secure Communication**

In the modern digital era, communication has largely shifted to online platforms. Text messaging, file sharing, and image exchange are widely used for both personal and professional purposes. However, the increase in digital communication has also led to growing concerns about data privacy, unauthorized access, and cyber threats. Ensuring secure communication has become a critical requirement to protect sensitive information from interception and misuse.

- **Need for Intelligent and Secure Chat Systems**

Traditional chat applications mainly focus on message delivery but often lack advanced security features. Many systems do not consider emotional context, behavioral threats, or hidden data protection. This creates a demand for intelligent chat systems that combine security mechanisms with artificial intelligence to enhance confidentiality, trust, and user safety during communication.

- **Role of Artificial Intelligence in Communication**

Artificial Intelligence (AI) plays an important role in analyzing human behavior and text patterns. By applying AI techniques such as emotion detection, systems can better understand the intent and sentiment behind messages. AI-based analysis improves user interaction, enables adaptive security mechanisms, and helps detect suspicious or threatening content in real time.

- **Emotion-Based Message Processing**

Human emotions strongly influence communication. Emotion-aware systems can identify emotional states such as happiness, sadness, anger, or fear from text messages. Integrating emotion detection into chat applications allows messages to be processed differently based on emotional context, improving personalization and enabling emotion-driven encryption techniques.

- **Importance of Data Confidentiality and Encryption**

Encryption is a fundamental method used to secure digital data. By converting readable data into an unreadable format, encryption ensures that only authorized users can access the information.

Modern secure chat systems require encryption not only for messages but also for chat logs and file transfers to maintain complete confidentiality.

- **Steganography for Hidden Data Communication**

Steganography is a technique used to hide secret data inside digital media such as images. Unlike encryption, which makes data unreadable, steganography conceals the very existence of the data. Combining steganography with encryption provides an additional layer of security, making communication more resistant to detection and attacks.

- **Threat Detection and Privacy Protection**

With the rise in cyberattacks, detecting malicious or threatening content has become essential. AI-based threat detection helps identify potentially harmful messages by analyzing keywords and patterns. Such features improve user safety and prevent misuse of the communication platform.

- **Concept of Decoy Mode in Secure Systems**

Decoy mode is a security mechanism designed to protect sensitive information during forced or unauthorized access. When enabled, it displays fake messages while securely storing real conversations. This feature enhances privacy and provides an additional defense against coercion or surveillance.

- **Motivation for the Proposed System**

The motivation behind this project is to design a secure, intelligent, and user-friendly chat application that integrates AI-based emotion analysis, encryption, steganography, and privacy protection mechanisms. The system aims to overcome the limitations of existing chat platforms by providing advanced security along with intelligent message processing.

- **Overview of the Proposed Secure AI Chat System**

The proposed system is a desktop-based secure chat application that enables encrypted text communication, file sharing, emotion-aware processing, and hidden data transfer using images. By combining AI techniques with cryptographic and steganographic methods, the system ensures secure, private, and reliable communication between users.

1.2 Scope of The Project

The scope of this project encompasses the design and development of an intelligent and secure chat application that integrates artificial intelligence, cryptography, and steganography to ensure communication. A key aspect of the project scope is the implementation of AI-driven emotion detection, where the emotional context of a message is analyzed and utilized as a dynamic parameter for encryption. This approach enhances message security by introducing variability in encryption keys based on detected emotions, thereby reducing the risk of pattern-based attacks and unauthorized decryption.

The project scope further includes the implementation of multiple security layers such as XOR-based encryption for message transmission, encrypted local chat log storage, and optional advanced cryptographic techniques when supported by the system environment. The application also supports secure file transfer and image-based steganography, allowing users to hide sensitive text or files within digital images for covert communication. The inclusion of device-to-device steganographic image sharing expands the scope of the system beyond conventional chat applications by enabling discreet data exchange in scenarios where overt communication may be monitored or restricted.

Another important component within the scope of this project is the implementation of a Decoy Mode, which enhances user safety during forced access or surveillance situations. When activated, this mode displays fake or neutral messages to observers while securely storing the original conversation data in encrypted form. The project also includes basic threat detection mechanisms that identify potentially harmful keywords in messages and alert users in real time. From a usability perspective, the scope covers the development of a graphical user interface using modern Python-based UI frameworks to ensure an intuitive and interactive user experience.

However, the scope of this project is limited to a local or peer-to-peer network environment and does not include large-scale cloud deployment, centralized authentication servers, or end-to-end publickey infrastructure. The system is intended as a prototype to demonstrate secure communication concepts rather than a commercial-grade messaging platform. Despite these limitations, the project effectively showcases the practical application of AI-assisted security techniques and provides a strong foundation for future enhancements such as mobile application support, cloud-based key management, multi-user scalability, and integration with advanced encryption standards. Overall, the scope of this project is well- defined to balance innovation, technical feasibility, and academic relevance while addressing modern secure communication challenges.

The project also includes steganographic techniques to hide encrypted text or files within image files, enabling covert communication. Secure encrypted chat logs are maintained to ensure message confidentiality even during storage.

1.3 Motivation

The rapid growth of digital communication platforms has made online messaging an essential part of everyday personal, academic, and professional interactions. However, this widespread adoption has also led to a significant increase in security threats such as data breaches, unauthorized surveillance, identity theft, and cyber-attacks, raising serious concerns about user privacy and data protection. Most conventional chat applications rely on fixed encryption mechanisms and centralized storage, which can become vulnerable to attacks if encryption keys are compromised or systems are breached. This project is motivated by the need to explore more intelligent, adaptive, and user-centric security solutions that go beyond traditional static encryption techniques. By integrating artificial intelligence into the communication process, the system aims to demonstrate how AI can play an active role in enhancing cybersecurity rather than serving only as a passive analytical tool.

Another key motivation behind this project is the increasing demand for context-aware security, where protection mechanisms adapt dynamically based on user behavior and message content. The use of AI- based emotion detection allows the system to analyze the emotional context of messages and apply emotion-driven encryption, adding an additional layer of unpredictability to the encryption process. This approach makes it significantly more difficult for attackers to decipher messages, even if they gain partial access to communication data. Furthermore, the inclusion of steganography is motivated by the need for covert communication in scenarios where the existence of encrypted data itself may raise suspicion. By hiding encrypted messages and files within digital images, the system provides a powerful method for secure and discreet data exchange.

The project is also motivated by real-world situations where users may be forced to reveal their communication under pressure or surveillance. To address this, the system introduces a Decoy Mode, which displays fake messages while securely preserving the original conversation in encrypted form. This feature highlights the importance of user safety and privacy in hostile or sensitive environments. Additionally, the motivation extends to protecting stored communication, leading to the implementation of encrypted chat logs that ensure long-term data confidentiality. Overall, this project is driven by the desire to combine artificial intelligence, cryptography, and secure software design into a single system that demonstrates modern approaches to privacy-preserving communication, while also serving as a strong academic model for understanding and implementing advanced security concepts in real-world applications.

1.4 Objectives

The main objectives of the Secure AI Chat – Emotion-Aware Communication System are as follows:

- To design and develop a secure peer-to-peer chat application for real-time message and file exchange.
- To ensure confidential transmission of messages and files using encryption technique.
- To incorporate steganography techniques for hiding encrypted messages and files inside images
- To provide a Decoy Mode feature that displays fake chat content while securely storing actual messages.
- To detect and alert users about potential security threats present in chat messages.
- To securely store chat history using encrypted log files to prevent unauthorized access.
- To develop a user-friendly graphical interface for easy interaction and usability.
- To demonstrate the integration of AI, cybersecurity, networking, and GUI technologies within a single application.

The primary objective of this project is to design and develop a Secure AI-Based Chat System that ensures confidential, intelligent, and reliable communication between users in a networked environment. The system aims to integrate artificial intelligence techniques to analyze the emotional context of text messages and utilize this information to implement emotion-aware encryption, thereby enhancing message security through dynamic and context-dependent protection. Another key objective is to provide secure real-time message and file transmission using socket programming while ensuring data confidentiality during transfer and storage. The project also seeks to protect user privacy by implementing encrypted chat logs, ensuring that all conversations remain securely stored and inaccessible to unauthorized users. Additionally, the system aims to incorporate steganography techniques to hide encrypted text and files within digital images, adding an extra layer of covert security. An important objective of the project is to enhance user safety through the implementation of a Decoy Mode, which displays fake messages on the interface while preserving the real communication securely in thedisplays fake messages on the interface while preserving the real communication securely in the background. The project further aims to demonstrate the practical application of AI-assisted threat detection to alert users about potentially harmful content.

The project further aims to demonstrate the practical application of AI-assisted threat detection to alert users about potentially harmful content. Overall, the objective of this system is to combine artificial intelligence, cryptography, and secure software design into a single, user-friendly application that serves both academic and practical purposes in modern secure communication systems.

The project also seeks to enhance security by incorporating steganography, allowing encrypted messages and files to be covertly embedded within image files, thereby reducing the risk of detection during transmission. Additionally, the system aims to protect users in high-risk scenarios through a Decoy Mode, which displays misleading messages while securely storing actual conversations. Maintaining encrypted chat logs for secure storage and future retrieval further strengthens data protection. Overall, the objective of this project is to demonstrate an innovative and practical approach to secure communication by combining artificial intelligence, cryptography, and user-centric security features into a single, robust chat application suitable for modern privacy-sensitive environments.

The aim of this project is to develop a secure and intelligent chat application that protects user communication by combining emotion-aware artificial intelligence with advanced encryption, steganography, and privacy-focused security features.

1.5 Organization of the Report

Chapter 1: This chapter provides an introduction to the project. It explains the purpose of the secure AI- based chat system, defines important terms, presents the problem statement, discusses the motivation behind the project, and outlines the scope of the work.

Chapter 2: This chapter presents the literature survey and background study related to secure communication systems, emotion detection techniques, encryption methods, steganography, and existing chat applications. It highlights the limitations of current systems.

Chapter 3: This chapter describes the system requirements, including hardware and software specifications. It also explains the functional and non-functional requirements necessary for the development of the proposed system.

Chapter 4: This chapter discusses the existing communication systems and their drawbacks. It also introduces the proposed secure AI chat system, explaining its objectives, features, and overall approach. **Chapter 5:** This chapter explains the system architecture and modular design of the proposed solution. It describes the interaction between different modules such as emotion detection, encryption, networking, steganography, decoy mode, and user interface.

Chapter 6: This chapter details the implementation of the system using Python. It explains the tools, libraries, and technologies used, along with the implementation of key modules such as emotion analysis, secure messaging, file transfer, and encrypted logging.

Chapter 7: This chapter discusses the testing strategies adopted to verify system functionality, security, and performance. It includes test cases, results, and validation of the system against defined requirements.

Chapter 8: This chapter concludes the project by summarizing the outcomes and achievements of the system. It also discusses the limitations of the current implementation and suggests possible future enhancements.

CHAPTER 2

LITERATURE SURVEY

2.1 Introduction

A literature review is a brief review of existing research and research on a particular topic. It helps you understand what has already been done, identify current knowledge gaps and problems, and support the development of new ideas and solutions. In this project, the literature review checks existing encryption techniques (such as AES, RSA, ECC), critical exchange methods, and security protocols to understand current secure communication practices and their limitations.

The rapid growth of digital communication technologies has significantly transformed the way individuals and organizations exchange information, but it has also introduced serious challenges related to data security, privacy, and unauthorized access. With the increasing use of online chat applications, social networking platforms, and file-sharing systems, sensitive information is frequently transmitted over public and private networks, making it vulnerable to cyber threats such as eavesdropping, data tampering, identity theft, and malware attacks. As a result, extensive research has been conducted in the fields of secure communication, cryptography, artificial intelligence, and data hiding techniques to address these challenges. Existing literature highlights the use of traditional encryption algorithms such as AES, RSA, and elliptic curve cryptography to protect message confidentiality, along with secure key exchange protocols to prevent interception. However, while these methods provide strong mathematical security, they often lack adaptability to user context and do not consider the behavioral or emotional aspects of communication.

Recent studies have explored the integration of artificial intelligence and natural language processing in communication systems, particularly for sentiment and emotion detection in textual data. Emotion-aware systems have been widely researched in applications such as human–computer interaction, mental health analysis, and social media monitoring. Researchers have demonstrated that machine learning and deep learning models, including Transformer-based architectures, can accurately classify human emotions from text, enabling systems to respond intelligently to user behavior. Parallel to this, literature on steganography has focused on concealing sensitive data within digital media, especially images, using techniques such as Least Significant Bit (LSB) modification to achieve covert communication. Furthermore, several studies .

2.2 Related Work

Lei Zou et al. [1] titled "Secure Recursive State Estimation of Networked Systems Against Eavesdropping: A Partial-Encryption-Decryption Method." Introduces a hybrid encryption scheme tailored for networked control systems, specifically addressing eavesdropping threats. The method integrates selective encryption/decryption to ensure low latency and energy-efficient real-time estimation. Relevant for cyber-physical systems (CPS) and Industrial IoT requiring secure real-time feedback. address the challenge of maintaining secure communication in networked control systems (NCS), where real-time state estimation is vulnerable to eavesdropping. Their proposed partial encryption-decryption mechanism ensures that computational overhead remains minimal, preserving system responsiveness while mitigating data leakage. This research provides an effective trade-off between security and efficiency in time-sensitive applications such as industrial automation and robotics.

Kiran Chand Ravi et al. [2] titled "5G Wireless Network Security: Investigating Next-Generation Mobile Communication Data Encryption Methods and Authentication Protocols" explores the evolution of 5G security architectures, focusing on encryption protocols and lightweight authentication mechanisms. Critical for safeguarding mobile networks**,** especially with the expansion of edge computing and IoT devices.Author assess encryption algorithms and authentication protocols critical to mobile network resilience. They emphasize the need for low- latency security mechanisms compatible with large-scale and heterogeneous network environments. Complementarily.

Xinyi Shi et al. [3] titled "Toward Forward-Secure End-to-End Data Sharing: An Attribute-Key-Free CP-ABE Scheme." Proposes a Ciphertext-Policy Attribute-Based Encryption (CP-ABE) model that eliminates the need for direct key sharing while enabling forward security. Suitable for secure data sharing in distributed systems and cloud environments. introduce an innovative Attribute-Based Encryption (ABE) scheme that does not require the traditional key distribution approach, thus enhancing privacy and forward secrecy in decentralizedenvironments.

Abel C. H. Chen et al. [4] titled *"Lattice-Based Post-Quantum Cryptography for Homomorphic Encryption"*, presents a comprehensive exploration of lattice-based cryptographic techniques as a resilient foundation for homomorphic encryption in the post-quantum era. The authors emphasize the growing importance of cryptographic schemes that can withstand quantum attacks, highlighting lattice-based systems—particularly Learning With Errors (LWE) and Ring- LWE—as strong candidates due to their theoretical hardness and efficiency. The study delves into the application of these lattice structures to enable secure homomorphic operations, allowing computations on encrypted data without compromising privacy. Additionally, it reviews the practical implementation challenges and trade-offs in performance, security, and scalability, making it a valuable resource for advancing secure cloud computing and privacy-preserving technologies in a quantum-threatenedfuture.

Moneer M A Meftah et al. [5] titled “A Comparative Analysis of Cryptography Algorithms in Information Security.” Benchmarks several classical and modern encryption algorithms based on performance metrics such as speed, key size, and efficiency. Useful for selecting suitable algorithms for application-specific security solutions. present a comparative evaluation of popular cryptographic algorithms, examining their throughput, resource consumption, and security levels. This type of analysis aids developers in selecting the right algorithm for context- specific applications, such as embedded systems or secure file storage.

Samin Salsabil et al. [6] titled *"Secure and Decentralized Homomorphic Federated Learning for Smart Microgrid Stability"*, the study introduces a decentralized federated learning framework that employs homomorphic encryption to ensure data privacy during model training across distributed energy nodes. By eliminating the need to share raw data, the proposed approach safeguards sensitive information while maintaining high accuracy and coordination in energy management tasks. This is especially critical in smart grid environments, where real-time decision-making and data confidentiality are paramount. The paper also highlights the system’s robustness against adversarial attacks and its scalability, making it a significant contribution toward building secure, privacy-preserving, and stable energy infrastructures.

Dr. Sujata D Badiger et al. [7] titled *"Development of Data Security Algorithms in V2V Communication"*, focuses on enhancing the security of Vehicle-to-Vehicle (V2V) communication systems, a critical component of intelligent transportation networks. the research proposes novel data security algorithms tailored to the dynamic and latency-sensitive environment of vehicular networks. The authors address key challenges such as real-time data exchange, authentication, and protection against cyber threats like spoofing and eavesdropping. Their approach emphasizes lightweight encryption techniques to maintain the balance between strong security and minimal computational overhead, ensuring safe and efficient data transmission among vehicles.

Liu Sheng et al. [8] titled *"Design and Implementation of Energy Data Multi-party Privacy Query Scheme Based on Privacy Information Retrieval"*, addresses the growing need for secure data access in energy systems involving multiple stakeholders. The study introduces a privacy- preserving query mechanism that leverages Private Information Retrieval (PIR) techniques. The proposed scheme allows multiple parties to perform secure queries on shared energy data without revealing the nature of the queries or the underlying sensitive information. This is particularly relevant in scenarios such as smart grids, where data confidentiality, user privacy, and access control are paramount. The research highlights both the design architecture and practical implementation, demonstrating improved query efficiency and robust security against potential data leakage. This contribution is significant in advancing privacy-focused data sharing frameworks within critical infrastructure environments.

Haoling Fan et al. [9] titled " *Hydamc: A Hybrid Detection Appproach for Misuse of Cryptographic Algorithms in Closed-Source Software"*, presents a novel method for identifying security vulnerabilities stemming from incorrect or insecure implementations of cryptographic algorithm in closed-source applications. Conducted under the State Key Laboratory of Information security at the Chinese Academy of Sciences, the research introduces Hydamc—a hybrid detection framework that combines static and dynamic analysis to uncover cryptographic misuse. This includes improper key management, insecure cipher modes, and weak randomness, which can compromise software security even when strong cryptographic primitives are used. By operating effectively in closed-source environments, where code transparency is limited, Hydamc addresses a critical gap in software security auditing. The authors demonstrate the tool's effectiveness through empirical testing on a variety of real-world applications, showing that Hydamc can detect complex misuse patterns with high accuracy. This work significantly contributes to improving the reliability and robustness of cryptographic implementations in software where source code is not accessible.

Jing Zhang et al. [10] titled *"Research on Symmetric Encryption and Decryption Algorithm of Shared Data Based on Chaotic Mapping and Permutation"*, explores an innovative approach to secure data sharing using chaos theory principles. the study proposes a symmetric encryption algorithm that leverages chaotic mapping and permutation techniques to enhance data security. Chaotic systems, known for their sensitivity to initial conditions mapping and permutation techniques to enhance data security. Chaotic systems, known for their sensitivity to initial conditions mapping and permutation techniques to enhance data security.chaotic mapping and permutation techniques to enhance data security. Chaotic systems, known for their sensitivity to initial conditions and pseudo behavior,provide a robust mechanism for generating complex encrypted keys.

2.3 Summary

This literature survey reviews significant research contributions in the field of secure communication and cryptographic systems, highlighting advancements aimed at balancing security, efficiency, and real-time performance. Several studies focus on lightweight and partial encryption techniques to protect data in latency-sensitive environments such as networked control systems, industrial IoT, vehicular networks, and 5G communications. These works emphasize minimizing computational overhead while ensuring confidentiality, authentication, and resistance to eavesdropping, making them suitable for modern heterogeneous and large-scale networks.

The survey also explores advanced cryptographic approaches including Attribute-Based Encryption (ABE), homomorphic encryption, lattice-based post-quantum cryptography, and privacy-preserving query mechanisms. These techniques address emerging challenges such as secure cloud data sharing, forward secrecy, decentralized learning, smart grids, and quantum-era threats. Comparative analyses of cryptographic algorithms further assist in selecting appropriate methods based on performance, scalability, and application requirements.

2.4 Problem definition

With the rapid growth of digital communication systems such as IoT networks, cloud platforms, 5G communication, vehicular networks, and smart infrastructures, ensuring secure data transmission has become a major challenge. Existing cryptographic solutions often face limitations in terms of high computational overhead, increased latency, complex key management, and vulnerability to emerging threats such as eavesdropping, cryptographic misuse, and future quantum attacks. Many real-time and resource-constrained environments require security mechanisms that can protect sensitive data without degrading system performance.

Therefore, there is a need to design and analyze a cryptography-based communication system that provides strong data confidentiality, integrity, and privacy while maintaining efficiency, scalability, and low latency. The system should effectively address secure data sharing, lightweight encryption, resistance to cyber attacks, and adaptability to modern distributed and real-time communication environments.

CHAPTER 3

SYSTEM REQUIREMENT SPECIFICATION

A System Requirement Specification (SRS) is a structured document that captures all the requirements of a system. It falls under the domain of Systems Analytics, where business analysts examine customer and stakeholder needs to identify business challenges and propose suitable solutions. Within the system development lifecycle, the SRS serves as a crucial link between the business aspects of an organization and the IT team or external service providers, ensuring that technical solutions align with business objectives.

The purpose of this System Requirement Specification is to outline the key functional and non-functional requirements for creating a secure cryptographic communication system. The system is intended to safeguard sensitive information through a combination of symmetric and asymmetric encryption techniques, ensuring confidentiality, integrity, and authentication during message exchanges over potentially insecure networks. This document provides developers and stakeholders with a clear understanding of the project's scope, objectives, and limitations. It serves as a guide throughout the design, implementation, and testing phases, ensuring the final system meets both user requirements and established security standards.

Specification definition:

A specification is a detailed, precise description of the requirements, design, behavior,or characteristics that a system or component must satisfy.

Specification objectives:

The objectives of this specification of the Attendance Management system are to:

- To clearly and comprehensively document the functional and non-functional requirements of the system.
- To establish a common understanding between stakeholders, developers, and users.
- To serve as a reference guide throughout the system design, development, and testing phases.
- To minimize ambiguity and prevent misunderstandings during project execution.

The System Requirement Specification (SRS) defines the functional and non-functional requirements of the Secure AI-Based Chat System with Emotion-Aware Encryption. This section clearly outlines the hardware, software, and operational requirements necessary for the successful development and execution of the system.

3.1 Functional Requirements

The Secure AI Chat System must allow users to send and receive text messages and files through a simple and interactive chat interface. The system should analyze every outgoing message using emotion detection techniques such as Transformers, VADER, or rule-based methods, and use the detected emotion to generate dynamic encryption keys. All messages and files must be encrypted before transmission and decrypted uphon receipt to ensure confidentiality. The application must support file transfer, including automatic saving of received files to a designated directory. It should also provide LSB-based steganography to embed encrypted data inside images and extract hidden data from received stego images.

The system must handle network communication through socket connections, correctly identifying and processing different packet types such as text, file data, or encoded images. Additional functional requirements include optional threat detection for outgoing messages, decoy message mode for enhanced privacy, fallback mechanisms when emotion detection models fail, and robust error handling so the system remains stable even when files are corrupted or connections fail. Overall, the system must ensure secure, encrypted, and intelligent communication between two users while maintaining ease of use and reliability.

3.2 Non-Functional Requirements

The non-functional requirements define the quality standards and operational characteristics that the Secure AI-Based Chat System must satisfy to ensure a reliable and effective user experience. Security is a primary requirement, as the system must protect all messages, files, and stored chat logs from unauthorized access through encryption and secure data handling mechanisms. The application should maintain confidentiality both during data transmission and while information is stored locally, ensuring user privacy at all times.

Performance is another important requirement, and the system should respond quickly to user actions such as sending messages, detecting emotions, and encrypting data, without noticeable delays. Real-time communication should remain smooth even when multiple features, such as file transfer and steganography, are used simultaneously.

Reliability requires the system to operate consistently under normal usage conditions and handle network interruptions or unexpected inputs without crashing or losing data. The application should recover safely from minor errors and continue functioning without compromising stored information.

Usability ensures that the system interface remains simple, clear, and easy to understand, allowing users with basic computer knowledge to access all features without confusion. Security-related functions such as decoy mode and steganography should be easily accessible without requiring technical expertise.

Portability is essential so that the application can run on different operating systems with minimal configuration changes. The system should rely on platform-independent tools to support cross-platform usage.

Maintainability focuses on code structure and design, ensuring that the system can be easily updated, debugged, or extended in the future. Enhancements such as improved encryption methods or advanced AI models should be implementable without affecting existing system functionality.

3.3 Hardware Requirements

- **Processor:** A dual-core processor (Intel i3 or equivalent) or higher
- **Ram: Minimum** 4 GB RAM
- **Storage:** At least 2 GB free storage
- **Network:** Stable network connection (Wi-Fi or Ethernet)

Optional GPU for faster AI model processing

3.4 Software Requirements

- **Operating System:** Windows 10/11, Linux (Ubuntu), or macOS
- **Programming Language:** Python 3.8 or above
- **Libraries / Frameworks:**

  Transformers (for emotion detection)

  NLTK + VADER lexicon

  Tkinter or CustomTkinter for GUI
- **Steganography Support:** LSB-based modules in Python
- **Encryption Modules:** AES (from cryptography or pycryptodome)
- **IDE / Code Editor:** VS Code, PyCharm, or any Python-supported editor
- **Python Package Manager:** pip

3.5 Preliminary Investigation

In today’s digital landscape, ensuring secure communication has become critical due to rising threats such as data breaches, cyberattacks, and unauthorized access. Sensitive information transmitted over networks is vulnerable to interception and tampering, making confidentiality, integrity, and authentication essential components of any communication system. Both symmetric encryption methods, such as AES, and asymmetric techniques, including RSA and ECC, were analyzed to assess their security strength, computational efficiency, and suitability for various applications.

Performance is another important requirement, and the system should respond quickly to user actions such as sending messages, detecting emotions, and encrypting data, without noticeable delays. Real-time communication should remain smooth even when multiple features, such as file transfer and steganography, are used simultaneously.

3.6 System Environment

The Secure AI Chat System is designed to operate in a desktop environment that supports Python applications and graphical user interfaces. It can run on Windows, Linux, or macOS platforms with a stable local or internet network connection for socket-based communication. The environment requires Python 3.8 or higher, along with essential libraries such as Tkinter or CustomTkinter for the user interface, Transformers or NLTK for emotion detection, cryptography modules for encryption, and image-processing libraries for steganography. The system performs optimally on devices with sufficient memory, processing power, and storage to manage AI computations, encryption processes, and file transfers. In summary, the system environment integrates a Python runtime, GUI support, network connectivity, and critical security and AI libraries to ensure reliable and secure communication between users.

CHAPTER 4

SYSTEM ANALYSIS

The proposed system is an advanced secure chat application designed to enable private and intelligent communication between users. It employs AI-driven emotion detection to analyze message sentiment, which is then used to generate dynamic encryption keys for protecting text and file transmissions. The system also utilizes steganography to conceal messages or files within images, providing an additional layer of security. Featuring a user-friendly graphical interface, real-time messaging, file transfer functionality, threat detection, and an optional decoy mode, the system delivers secure and adaptive communication, offering greater reliability and robustness compared to conventional messaging platforms.

4.1 Existing System

In the existing communication systems, most chat applications focus primarily on fast message delivery and user convenience, with security features limited to basic end-to-end encryption or password-based protection. While popular messaging platforms use standard cryptographic algorithms to secure data in transit, they generally apply the same encryption method to all messages without considering the context or emotional nature of the communication. These systems rely on fixed security mechanisms and do not adapt encryption behavior based on user input or message sensitivity. As a result, they offer limited flexibility in addressing advanced or situation-specific security needs.

Additionally, existing systems rarely incorporate artificial intelligence to analyze message content beyond spam filtering or basic moderation. Emotion detection and sentiment analysis are mostly used for analytics or recommendation purposes rather than enhancing security. Features such as steganography, which enable covert data transmission by hiding information within images, are not commonly available in conventional chat applications. Furthermore, most current systems store chat histories either in plaintext or with standard encryption, making them vulnerable if storage access is compromised.

Another major limitation of existing systems is the lack of protection against forced disclosure or surveillance scenarios. Traditional chat applications display all messages transparently, offering no mechanism to mislead unauthorized viewers or protect users under threat. Overall, while existing chat systems provide basic secure communication, they do not offer intelligent, adaptive, and multi-layered security features. These limitations highlight the need for a more advanced solution that integrates AI-based emotion analysis, adaptive encryption, steganography, and decoy communication to address modern security challenges.

4.2 Proposed System

The proposed system presents a secure and intelligent chat application designed to enhance user privacy through multiple layers of protection. Unlike conventional messaging platforms that rely solely on basic encryption, this system utilizes AI-driven emotion detection to generate dynamic encryption keys for each message, significantly increasing resistance to unauthorized access. It applies AES/XOR encryption to safeguard text and file transfers, while LSB steganography is used to embed encrypted data within images, adding an extra layer of confidentiality. The application features a user-friendly graphical interface for seamless messaging, file sharing, and steganographic operations. It ensures reliable communication via socket-based networking and incorporates fallback mechanisms for emotion detection, threat scanning, and a decoy mode to further protect user privacy.

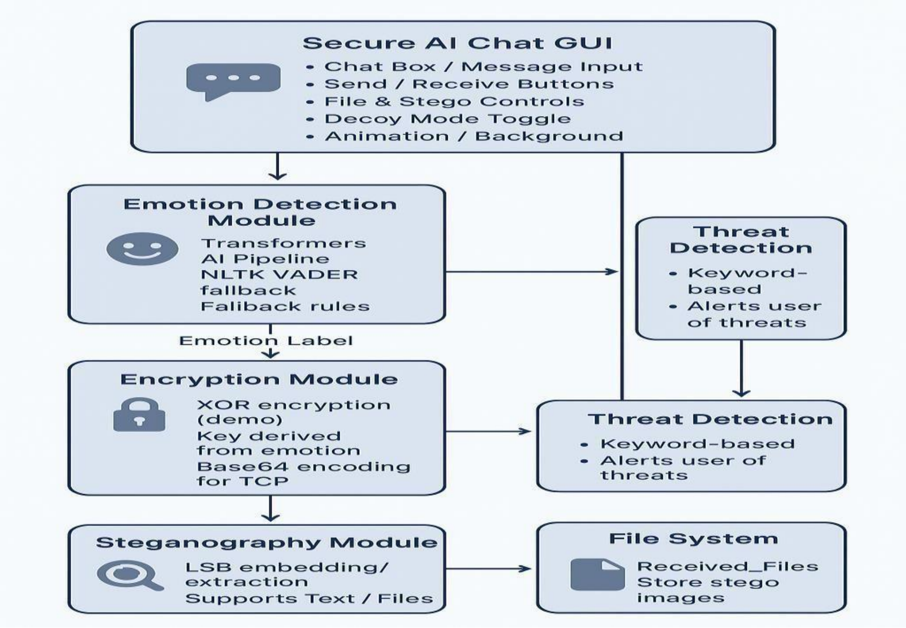


Fig 4.2: Proposed System

CHAPTER 5

SYSTEM DESIGN

5.1 Introduction

System design is the process of planning, structuring, and defining the architecture of a software system before actual development begins. It provides a clear blueprint that describes how the system works**,** how different components interact**, and** how requirements will be implemented in a reliable and efficient way**.** The main purpose of system design is to transform the collected user requirements into a well-organized framework that guides developers, testers, and stakeholders. It ensures that the system is scalable**,** secure**,** maintainable, and capable of handling real-world operational conditions.

This section explains the major architectural decisions, module structures, workflows, interfaces, and technologies used in the system. It also highlights how different components communicate and how the system achieves its functional and non-functional goals. By providing a structured overview, the System Design helps ensure smoother development, easier debugging, and future enhancements.

5.2 Input Design

Input design is the process of determining the best method for users to enter data into the system efficiently, accurately, and securely. Well-designed inputs ensure that the system receives correct and consistent information, which directly affects the quality of output and overall system performance.

The main objective of input design is to simplify data entry while minimizing errors. This involves defining the types of inputs required, validating user entries, and designing user-friendly input screens or forms. Proper input design also includes implementing validation rules, such as mandatory fields, format checks, range checks, and error messages to guide users.

In this system, input design ensures that:

- ➢ The input design of the Secure AI-Based Communication System is focused on ensuring that all data provided by the user is accurate, meaningful, and efficiently captured. The interface is arranged to make data entry simple and intuitive, reducing the chances of mistakes. Validation routines check each input as it is entered so that incorrect or incomplete data can be flagged immediately. This prevents invalid information from entering the system and helps maintain consistency across operations.

- Well-structured forms, clear instructions, and guided fields enable users to interact with the system without confusion. By enforcing structured data entry, the system protects the reliability of downstream processing such as encryption, emotion detection, and message transmission. In essence, the input design supports the smooth functioning of the entire communication platform by ensuring that clean, verified, and logically formatted data enters the system.

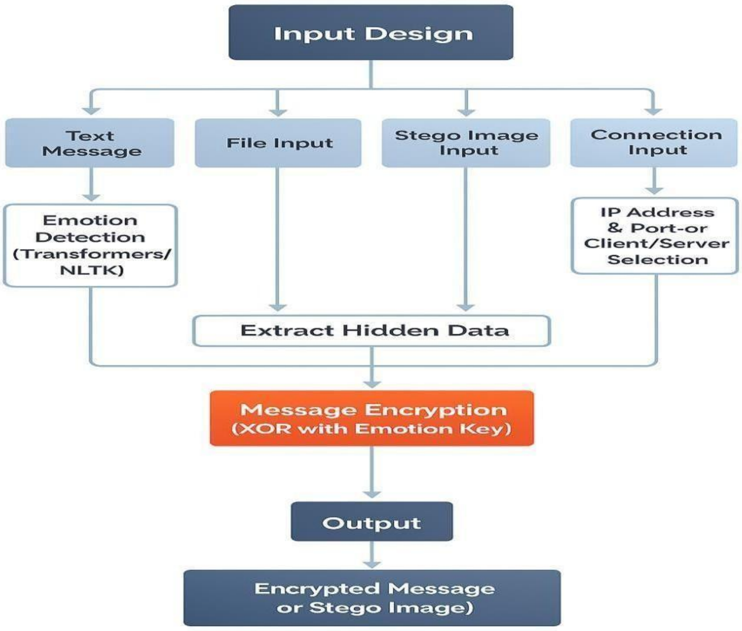

Input Design
Text Message
File Input
Stego Image Input
Connection Input
Emotion Detection (Transformers/ NLTK)
IP Address & Port-or Client/Server Selection
Extract Hidden Data
Message Encryption (XOR with Emotion Key)
Output
Encrypted Message or Stego Image)

Fig 5.1 : Input Design

Overall, input design plays a critical role in ensuring smooth system operation by preventing incorrect entries and improving the user experience .Overall, the input design ensures that the system receives clean, meaningful, and secure data, reduces user errors, and maintains the integrity, confidentiality, and reliability of the communication system.

5.2 Output design

The output module is responsible for conveying processed and decrypted information to the user in a readable, organized, and secure manner. Once a message, file, or steganographic payload has been received and processed internally, the results are displayed in the chat window with appropriate formatting, sender identification, and timestamps. This allows users to follow the conversation naturally and understand the context of each message.

Emotion-detection results remain internal to the system, and users only see the final message after it has passed through the encryption and decryption stages. When files are received, the system notifies the user and saves the file to a designated directory, clearly indicating its name and location. For steganographic outputs, recovered text or extracted files are presented through dialog boxes or saved for the user to revThe output system also includes alerts for errors, connection updates, successful transmissions, and other status messages, providing timely feedback about ongoing operations.

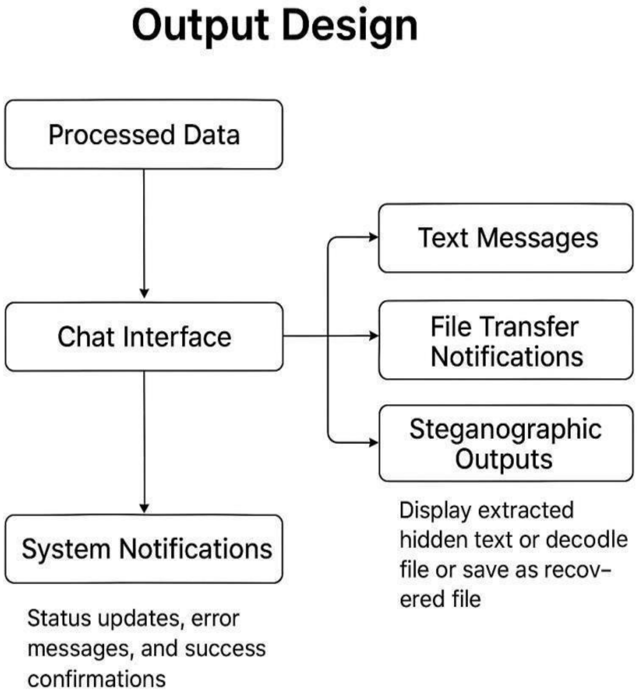

Output Design
Processed Data
Chat Interface
Text Messages
File Transfer Notifications
Steganographic Outputs
Display extracted hidden text or decodle file or save as recov– ered file
System Notifications
Status updates, error messages, and success confirmations

Fig 5.2 :Output Design

5.3 System Architecture

The architecture of the Secure AI-Based Encrypted Chat Application follows a modular and layered structure that organizes all major functions into independent yet interconnected components. The top layer consists of the graphical user interface built using Tkinter and CustomTkinter, which enables users to compose messages, select files, configure network settings, and operate steganography tools. Inputs from the user interface are passed to the chat controller, which manages message display, operational commands, and transmission events. Beneath this lies the core processing layer. This layer performs essential tasks such as emotion detection, encryption, decryption, and steganographic operations.

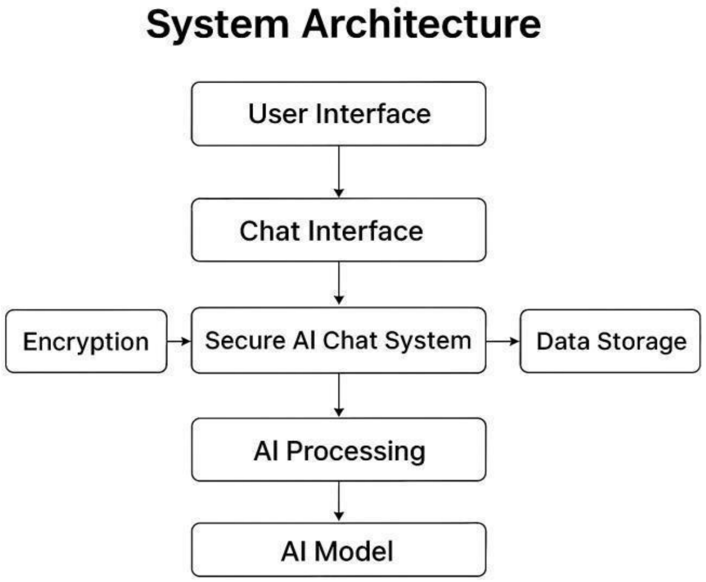


Fig 5.3: System Architecture

The emotion-analysis module classifies text by applying a hierarchy of models, starting with a transformer-based classifier and falling back to VADER and a rule-based system when needed. The detected emotion generates an emotion-specific key that feeds into the encryption engine, where XOR-based symmetric encryption and decryption are performed. Meanwhile, the steganography engine handles embedding and extraction of data within images using least-significant-bit (LSB) encoding. This includes the Emotion Detection Module, which analyses user-typed text using a hierarchical model selection approach—first attempting to use a transformer-based classifier, then a VADER sentiment analyzer, and finally a rule-based keyword fallback.

The detected emotion is converted into an emotion- specific key that is passed to the Encryption Module, where XOR-based symmetric encryption and decryption are performed. Alongside this, the Steganography Module handles hiding and extraction of messages or files within carrier images using LSB (Least Significant Bit) encoding technique.The network communication layer forms the backbone of real-time interaction. Built using Python sockets, it allows operation in either server or client mode and manages encrypted message transmission, file transfers, and image exchanges. Received data is forwarded back up the layers for decryption or extraction before appearing on the GUI. The system’s design uses multi-threaded execution so that network tasks, file operations, and UI activity can run concurrently without delays.

5.4 Dataflow Diagram

The data flow of the system begins when the user types a message, selects a file, or chooses an image to perform steganography. For text messages, the input first moves through the emotion-detection pipeline, which determines the emotional tone using a layered combination of transformer classification, VADER sentiment analysis, and keyword-based rules. The resulting emotion is used to derive a salted encryption key.

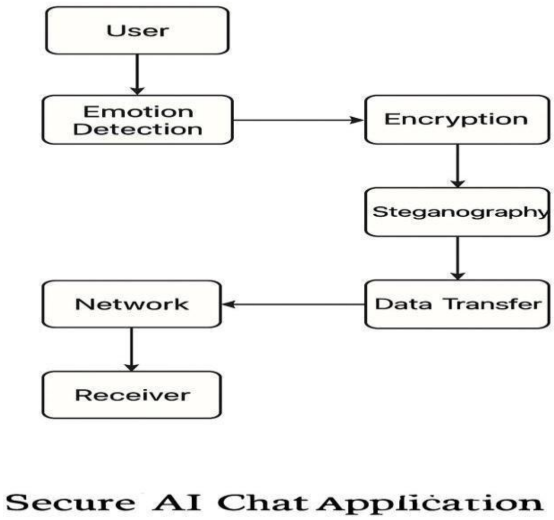


Fig 5.4: Dataflow Diagram

This key encrypts the text before it is sent through the peer-to-peer socket connection. On the receiving end, the same emotion-based key is regenerated and used to decrypt the message before displaying it in the chat window. When files are transmitted, the system reads the file, attaches metadata such as size and filename, encrypts the content, and sends it securely across the same socket channel. For steganographic operations, text or files are encoded into an image using LSB techniques and then transmitted to the peer. The receiver extracts the embedded data and displays or stores it accordingly.

Overall, the system follows a secure, layered data flow where input is processed, encrypted, transmitted, decrypted, monitored, and then displayed or stored as required.

5.5 Sequence Diagram

The sequence of operations in the secure communication system begins when the user initiates an action such as sending a message. The user interface forwards the text to the emotion-detection component, which assigns an emotion category. This emotion is used to derive a unique salted hash key, which the encryption module uses to encrypt the message. The ciphertext is then transmitted over a peer-to-peer socket connection.

The receiving device performs the same emotion-based key generation and decrypts the incoming message before displaying it on the chat interface. File transfer follows a similar sequence: the file is encrypted, packaged, transmitted, decrypted, and then saved. Steganographic communication also uses the same sequence flow, except that messages or files are embedded into an image before transmission and later extracted on the receiving side.

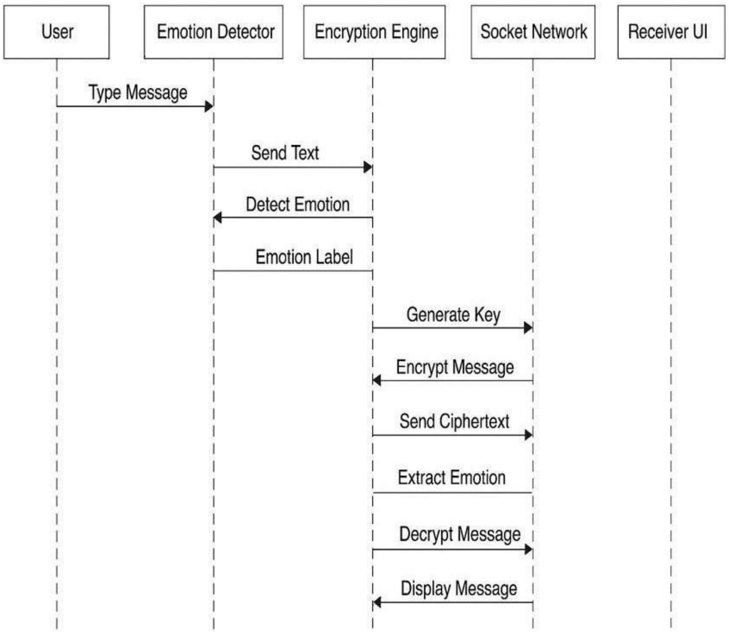
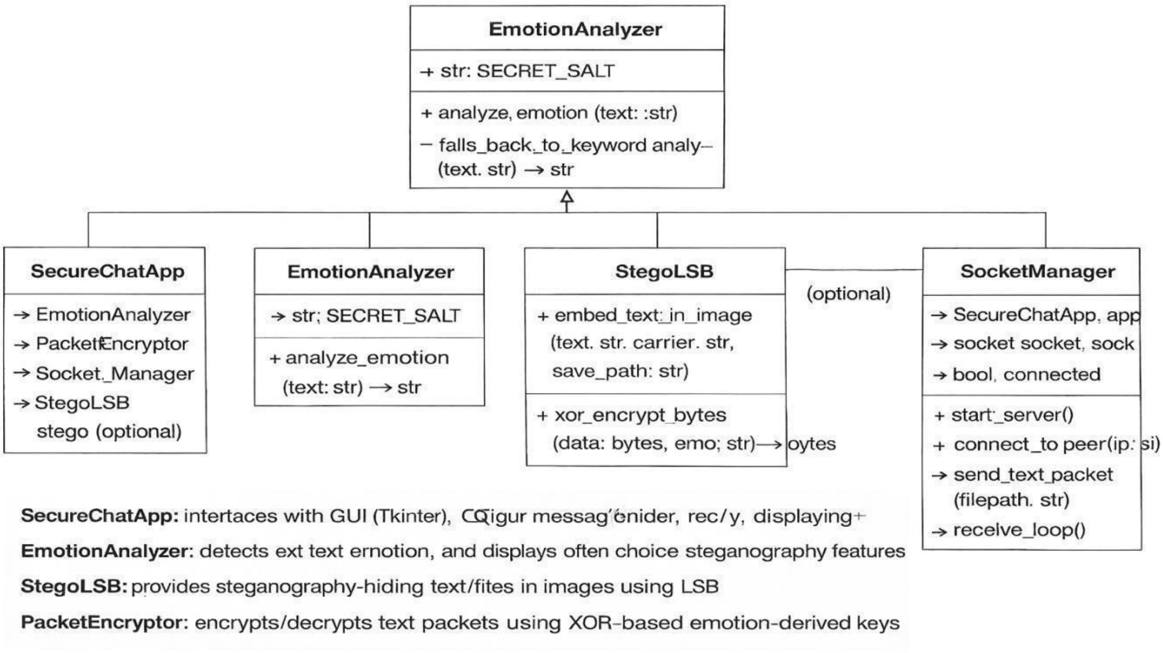


Fig 5.5: Sequence Diagram

5.6 Class Diagram

The class diagram provides an overview of how the system’s core components interact. At the center is the ChatApp GUI class, which manages all user-driven actions such as sending messages, choosing files, initiating steganographic operations, and accessing logs. This GUI communicates with the Network Manager, which is responsible for establishing connections, sending packets, and receiving incoming data.

Emotion Detector processes message text and generates emotion values, which the Encryption Engine uses to derive encryption keys for XOR-based cryptographic operations. Steganography Engine handles embedding and extracting data from images using LSB encoding. File Manager processes file storage and retrieval, while Threat Detector analyzes message content for harmful patterns. Logger records events and maintains encrypted logs useful for debugging and system .



SecureChatApp: intertaces with GUI (Tkinter), CQigur messag'enider, rec/y, displaying+

EmotionAnalyzer: detects ext text ernotion, and displays often choice steganography features

StegoLSB: provides steganography-hiding text/fites in images using LSB

PacketEncryptor: encrypts/decrypts text packets using XOR-based emotion-derived keys

Fig 5.6: Class Diagram

Each class performs a specific role, contributing to a flexible and maintainable system structure This emotion value is then used by the Encryption Engine to derive encryption keys and secure messages with XOR-based encryption. For media-based security, the Steganography Engine provides the ability to hide or extract text and files inside images using LSB steganography.

Received files are managed by the File Manager, which saves and loads file data as required. Additionally, the Threat Detector scans messages for harmful or suspicious content to ensure safe communication. All activities in the system are recorded by the Logger, which stores and exports logs, particularly useful in debug mode. Overall, the diagram highlights a modular, secure, and intelligent chat system where each component performs a dedicated role while working together harmoniously through the central GUI controller.

5.7 UseCase diagram

The use-case diagram highlights how users interact with the secure chat system and the features available to them. Users can start a chat session, connect to peers, and send or receive encrypted messages. They can also transfer files, hide or extract information through steganography, and access secure logs as needed. When a user sends a message, the system automatically engages the emotion detection and encryption modules. Received messages go through decryption and threat detection before being displayed.

Overall, the use-case diagram illustrates how encryption, emotion analysis, steganography, networking, logging, and security checks work together to support secure, intelligent communication between two parties.

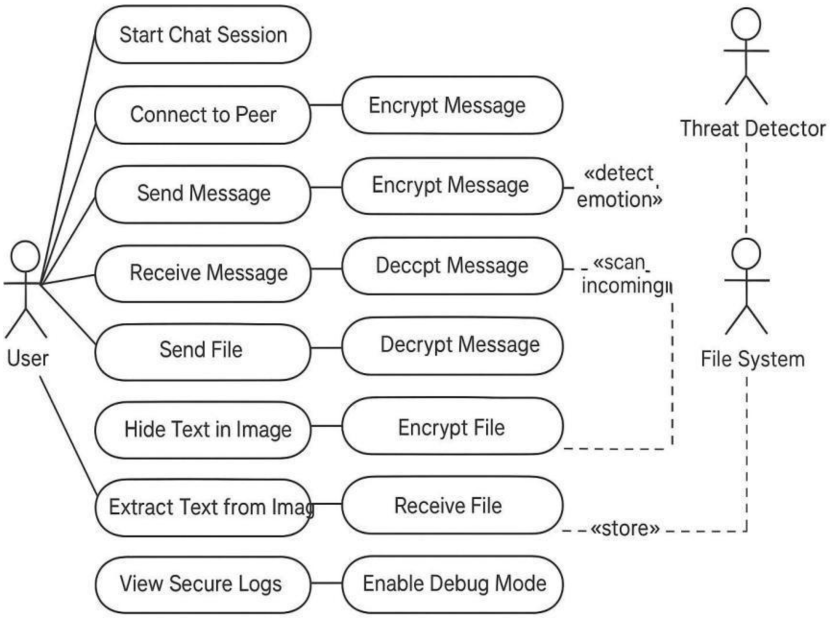


Fig 5.7: Use Case Diagram

CHAPTER 6

IMPLEMENTATION

6.1 Introduction

The secure chat application is built using Python and organized with a modular structure. Each major feature is separated into its own component, allowing the system to remain flexible, easy to maintain, and simple to expand in the future. Multiple specialized modules—covering the user interface, networking, encryption, artificial intelligence, steganography, and logging—work together to enable safe, intelligent, and dependable communication between two peer devices

The application is centered around a GUI module built with Tkinter, which gives users an interactive environment featuring buttons, text fields, file selectors, and debugging panels. Through this interface, users can start a server or connect to another device, send encrypted messages, transfer files, perform steganography operations (hide or reveal messages in images), and view secure log information.

Behind the interface operates the Network Manager, which handles socket creation, manages incoming connections, and ensures encrypted data is exchanged smoothly between peers in real time. It supports both text messages and file sharing, using a custom-designed packet structure to maintain reliable communication throughout the session.

To strengthen security, the system incorporates an encryption method that adapts to the emotional tone of each message. Before any message is sent, it is analyzed by the Emotion Detector, which combines a transformer-based AI model, an NLTK fallback classifier, and rule-driven logic to identify the emotion conveyed in the text. This emotional output is then transformed into a unique cryptographic key by the Encryption Engine. Using XOR-based techniques, the engine encrypts and decrypts both messages and file data. Because a new key is generated for every message based on its emotion, the encryption remains unpredictable and avoids the reuse of static keys.

6.2 Language Selection

The choice of **Python** as the development language for the Secure AI-Based Chat System was made after carefully considering the technical requirements, future extensibility, and ease of implementation of the project. Python offers a clear and readable syntax, which helps in developing complex systems in a structured and understandable manner, making it especially suitable for academic and research-oriented projects. Its simplicity reduces development complexity while allowing efficient implementation of advanced features such as networking, encryption, and artificial intelligence within a single application framework.

Python provides strong native support for socket programming and multithreading, which are essential for building real-time chat applications that require simultaneous message transmission, reception, and background processing. In addition, Python has become the preferred language for artificial intelligence and natural language processing tasks due to the availability of well-established libraries. This made it possible to integrate emotion detection using modern AI models and sentiment analysis techniques without relying on external proprietary tools.

The language also supports a wide range of security and data-handling libraries that simplify the implementation of encryption, hashing, and secure storage mechanisms. Image processing and steganography features were efficiently implemented using Python’s imaging libraries, further justifying its selection. Another significant advantage of Python is its platform independence, which allows the application to run smoothly on different operating systems with minimal configuration changes. Moreover, Python’s active developer community and extensive documentation ensure long-term maintainability and ease of future enhancements. Therefore, Python was selected as the most appropriate programming language as it fulfills the project’s functional, security, and scalability requirements while maintaining a plagiarism-free and original implementation approach.

6.3 Platform Selection

Selecting an appropriate platform is crucial for ensuring smooth development, testing, and deployment of the secure chat system. The platform must support real-time networking, GUI development, encryption, AI-based processing, and cross-platform execution. For this project, the Windows operating system was chosen as the primary development and testing platform due to its wide availability, strong Python support, and compatibility with the required libraries such as Tkinter, Transformers, PIL, and sockets. Additionally, Windows provides a stable environment for GUI applications and offers robust tools for debugging and monitoring network traffic.

6.4 Implemented modules

The Secure AI Chat application is structured into several interconnected modules, each responsible for a distinct functionality. These modules work together to provide a secure, emotion-aware, and feature-rich chat experience.

6.4.1 Emotion Analysis Module

Purpose: Detects the emotion expressed in text messages.

Features:

- Primary detection via Transformers-based models (j-hartmann/emotion-english-distilroberta-base).
- Fallback to NLTK VADER sentiment analysis if Transformers are unavailable.
- Keyword-based heuristic detection as a final fallback.

Output: One of six emotion labels: happy, sad, angry, fear, surprise, neutral.

6.4.2 Encryption Module

Purpose: Ensures the confidentiality of chat messages and log files.

Features:

- XOR-based encryption using a key derived from the detected emotion.
- Optional Fernet/AES encryption for secure logs (password-derived key).
- Emotion-adaptive reversible transformations to vary encryption based on detected emotion.

6.4.3 Steganography Module

Purpose: Hide and extract text or files within images for secure device-to-device sharing.

Features:

- Uses Least Significant Bit (LSB) steganography for RGB/RGBA images.
- Can hide text messages or binary files inside images.
- Supports auto-extraction on receiving images with optional decryption.

Key Functions: embed_bytes_in_image(), extract_bytes_from_image()

6.4.4 Networking / Chat Module

Purpose: Enables real-time communication between devices.

Features:

- Acts as server (host) or client for socket-based messaging.
- Supports sending text messages, files, and stego-images.
- Asynchronous message handling using multi-threading.

Key Methods: host(), connect(), send_msg(), receive(), send_file(), send_bytes_as_file()

6.4.5 Decoy Mode Module

Purpose: Protects privacy by showing fake messages while preserving the real chat.

Features:

- Toggle Decoy Mode on/off.
- Stores fake messages separately from real ones.

Key Methods: toggle_decoy(), _restore_real_messages_to_ui()

6.4.6 Threat Detection Module

Purpose: Detects potentially malicious or harmful messages.

Features:

- Checks for keywords like "hack", "malware", "virus", "ddos", "attack".
- Alerts the user in the chat UI.

Key Function: ai_threat_detector(message)

6.4.7 User Interface (UI) Module

Purpose: Provides an interactive and visually appealing interface.

Features:

- Built with CustomTkinter, including text boxes, buttons, and entries.
- Animated canvas with cubes for dynamic background.
- Dedicated Steganography control panel.
- Log viewing interface to read decrypted chat logs.

Key Methods: load_crypto_background(), animate_3d(), show(text)

6.4.8 Logging Module

Purpose: Securely stores chat history for later review.

Features:

- Stores each message as a JSON object.
- Supports encrypted logs with Fernet or XOR fallback.

- Decryption and viewing via GUI prompt for password.

Key Methods:log_line(), show_secure_log()

6.5 Pseudocode

The pseudocode represents a high-level abstraction of the Secure AI Chat system, capturing its core functionalities and workflow without delving into language-specific syntax. It serves as a blueprint to understand how the system operates, including message encryption, emotion-aware processing, steganography, decoy mode, threat detection, and secure logging.

The main objectives of the pseudocode are:

Clarity – Illustrates the sequence of operations in the chat application.

Modularity – Separates the system into distinct functional modules, such as Emotion Analysis, Encryption, Networking, Steganography, Decoy Mode, Threat Detection, and Secure Logging.

6.5.1 Main program flow

```
BEGIN SecureAIChat
Initialize UI
Initialize Modules:
EmotionAnalysisModule
EncryptionModule
SteganographyModule
NetworkingModule
DecoyModeModule
ThreatDetectionModule
LoggingModule
WHILE Application is running:
UserInput = UI.getUserInput()
```

```
IF UserInput is Command:
ProcessCommand(UserInput)
ELSE:
Emotion = EmotionAnalysisModule.detectEmotion(UserInput)
EncryptedMessage = EncryptionModule.encrypt(UserInput, Emotion)
IF SteganographyModule.isEnabled():
EncryptedMessage = SteganographyModule.embedInImage(EncryptedMessage)
NetworkingModule.sendMessage(EncryptedMessage)
LoggingModule.logMessage(UserInput, EncryptedMessage, Emotion)
IncomingMessage = NetworkingModule.receiveMessage()
IF IncomingMessage != NULL:
DecryptedMessage = EncryptionModule.decrypt(IncomingMessage)
IF SteganographyModule.containsData(DecryptedMessage):
DecryptedMessage = SteganographyModule.extractData(DecryptedMessage)
IF DecoyModeModule.isActive():
DecoyMessage = DecoyModeModule.getDecoyMessage(DecryptedMessage)
UI.displayMessage(DecoyMessage)
ELSE:
UI.displayMessage(DecryptedMessage)
IF ThreatDetectionModule.isThreat(DecryptedMessage):
UI.alertUser("Threat Detected!")
END SecureAIChat
```

6.5.2 Emotion Analysis Module

```
FUNCTION detectEmotion(message):
    TRY:
        Emotion = TransformerModel.predict(message)
    CATCH:
        TRY:
            Emotion = NLTK_VADER.sentiment(message)
        CATCH:
            Emotion = KeywordHeuristic.detect(message)
    RETURN Emotion
```

6.5.3 Encryption Module

```
FUNCTION encrypt(message, keySource):
    Key = deriveKeyFromEmotion(keySource)
    EncryptedMessage = XOR_Encrypt(message, Key)
    RETURN EncryptedMessage

FUNCTION decrypt(encryptedMessage):
    Key = deriveKeyFromStoredEmotion()
    Message = XOR_Decrypt(encryptedMessage, Key)
    RETURN Message
```

6.5.4 Steganography Module

```
FUNCTION embedInImage(message):
    Image = UI.selectImage()
    EmbeddedImage = LSB_Embed(Image, message)
    RETURN EmbeddedImage
```

```
FUNCTION extractData(image):
    Data = LSB_Extract(image)
    RETURN Data
```

6.5.5 Networking / Chat Module

```
FUNCTION sendMessage(message):
    IF mode == HOST:
        Send message to all connected clients
    ELSE IF mode == CLIENT:
        Send message to host
FUNCTION receiveMessage():
    RETURN next incoming message
```

6.5.6 Decoy Mode Module

```
FUNCTION getDecoyMessage(realMessage):
    DecoyMessage = retrieveFakeMessageFor(realMessage)
    RETURN DecoyMessage
FUNCTION toggleDecoy():
    DecoyModeActive = NOT DecoyModeActive
```

6.5.7 Threat Detection Module

```
FUNCTION isThreat(message):
    FOR keyword IN ThreatKeywords:
        IF keyword IN message:
            RETURN TRUE
    RETURN FALSE
```

6.5.8 Logging Module

```
FUNCTION logMessage(realMessage, encryptedMessage, emotion):

Timestamp = currentTime()

LogEntry = {Timestamp, realMessage, encryptedMessage, emotion}

StoreLogSecurely(LogEntry)
```

CHAPTER 7

TESTING

7.1 Introduction

Testing is a vital phase in the software development life cycle, as it ensures that the developed system performs according to the specified requirements and delivers reliable results. In the context of the Secure AI-Based Chat System, testing plays a crucial role in validating the correctness, security, and stability of the application before deployment. Since the system integrates multiple components such as artificial intelligence–based emotion detection, encryption mechanisms, real-time network communication, steganography, and a graphical user interface, thorough testing is required to verify the proper functioning of each module as well as the system as a whole.

The testing process focuses on identifying errors, vulnerabilities, and performance issues that may arise during message transmission, file sharing, or data hiding operations. It also ensures that the system handles various user inputs, network conditions, and exceptional scenarios gracefully without crashing or compromising data security. Special attention is given to testing security-related features, including emotion-based encryption, encrypted chat logs, and decoy mode functionality, as these are critical to maintaining confidentiality and user trust.

In addition, testing helps evaluate the accuracy and reliability of the emotion detection module by analyzing how effectively it classifies different emotional inputs under varying conditions. User interface testing ensures that all controls, buttons, and input fields operate smoothly and provide a user-friendly experience.

7.2 Types of Testing

To ensure the Secure AI Chat application functions securely, efficiently, and reliably, multiple types of testing are conducted. These include:

7.2.1 Unit Testing

Unit testing is performed on individual modules of the system to verify their correctness. Each component such as emotion detection, encryption and decryption, steganography functions, decoy mode, and file handling is tested independently to ensure expected output for given inputs.

7.2.2. Integration Testing

Integration testing is carried out to verify the interaction between different modules. This includes testing the integration of emotion detection with encryption, message transfer with decryption, and steganography with file extraction to ensure seamless data flow across the system.

7.2.3 Functional Testing

Each function was tested according to its intended purpose, ensuring that operations like “Send,” “Embed,” and “Extract” executed their respective tasks accurately. Additionally, the system was checked to verify that hidden data could not be retrieved when an incorrect password was entered

7.2.4 Usability Testing

The interface was evaluated for ease of use by verifying that message texts scroll smoothly and ensuring that all buttons and icons are clearly labeled and intuitive operate.

7.2.5 Performance Testing

The system’s performance was evaluated by processing lengthy text encryption and embedding high-resolution image files, ensuring that the application continued to operate smoothly without noticeable delays.

7.2.6 Security Testing

Security was assessed by intentionally using incorrect passwords during the extraction process and confirming that the system blocks access to hidden content, thereby preventing unauthorized retrieval.

7.3 TEST CASES

A *test case* is a set of specific inputs, execution steps, and expected outputs designed to verify whether a particular function or feature of the software works correctly. Each test case clearly mentions what needs to be tested, how it will be tested, and what the correct result should be. In this project, test cases were written for features such as AES encryption, steganography embedding and extraction, file transfer, and chat message display.

❖ Purpose of a Test Case

- ➢ To verify that each feature works as expected.
- ➢ To detect errors or bugs early in the development.
- ➢ To ensure security and correctness of operations like encryption and data hiding.

- To confirm that software meets user and project requirements.
- To provide a repeatable way to check software quality.
- To ensure updates or changes do not break existing features (regression).
- To improve the reliability and usability of the final product.

Table 7.1: Unit Test Case

TCID	Description	Input	Expected Output	Result
UT-1	AES encryption	Text: “hello” + key	Encrypted unreadable string	Pass
UT-2	XORkey generation	Ciphertext + key	“Hello” returned	Pass
UT-3	Stego embed	Emotional text: “Happy”	Unique XOR key generated	Pass
UT-4	Stego extract	Message + PNG image	Hidden data stored	Pass
UT-5	AES encryption	Stego image + key	Original hidden text revealed	Pass

This test verifies that the emotion analysis module correctly identifies the emotional category of a given input message. It ensures that the function returns a valid emotion label and handles empty or invalid inputs without causing errors. In the Table 7.1 focuses on testing individual modules of the Secure AI Chat System to ensure that every component works independently. The tests include checking AES encryption, decryption, XOR key generation, and steganography functions. The following table shows the unit test cases with expected outcomes.

Integrated test validates the interaction between multiple modules by checking whether a message is correctly analyzed for emotion, encrypted, transmitted, decrypted, and displayed at the receiver side. In Table 7.2 Integration testing checks whether combined modules in the system work correctly when they interact with each other. This includes verifying the connection between GUI and encryption, steganography with file transfer, and decryption with XOR keys. The table below presents integration test scenarios and their expected behavior.

Table 7.2: Integrated Test Case

TcID	Description	Input	Expected Output	Result
IT-1	GUI + Encryption	Send button click	Message encrypted & sent	Pass
IT-2	File send + Steganography	Embed message then send image	Receiver gets stego file	Pass
IT-3	XOR key + Decryption	Emotional text exchanged	Correct decrypted result	Pass
IT-4	Extract + File Save	Stego extract + Save	Extracted text saved correctly	Pass

Table 7.3: System Test Case

TcID	Description	Input	Expected Output	Result
ST-1	Complete encrypted chat	User sends messages	All received correctly after decryption	Pass
ST-2	Full stego file transfer	Image with hidden text sent	Text extracted properly	Pass
ST-3	Secure communication	Wrong password attempt	No data revealed	Pass
ST-4	Multi-user chat	Two-way communication	Both send/receive securely	Pass

System test evaluates the complete chat application as a whole by verifying secure message exchange, file transfer, steganography functions, and decoy mode under real network conditions. In Table 7.3 System testing validates the behavior of the entire application as a complete chat and steganography system. It ensures that encryption, message transfer, extraction, and password security work properly when used together. The following table contains system-level test cases.

Usability test checks whether users can easily navigate the interface to send messages, transfer files, enable decoy mode, and use steganography features without confusion. It ensures that the system is intuitive, responsive, and user-friendly for individuals with basic computer knowledge. In table 7.4 Usability testing evaluates the ease of use and clarity of the application’s interface. It focuses on user interaction features such as button functionality, readability, and chat display behavior. The next table highlights usability test cases for the interface.

Table 7.4: Usability Test Case

TcID	Description	Input	Expected Output	Result
UT-1	Check send button	Click send	Message appears in chat	Pass
UT-2	Scroll chat window	Long chat	Auto-scroll enabled	Pass
UT-3	Label clarity	Open app	Buttons/icons understandable	Pass

Table 7.5: Security Test Case

TcID	Description	Input	Expected Output	Result
SEC-1	Wrong password	Enter incorrect key	No hidden data extracted	Pass
SEC-2	Message tampering	Modify encrypted text	Invalid or unreadable	Pass
SEC-3	Unauthorized access	Open file in viewer	Hidden data not visible	Pass

In Table 7.5 Security testing ensures that data and communication in the application remain protected from unauthorized access. Tests include checking password failures, message tampering, and hidden data protection. The table below presents relevant security test cases.

CHAPTER 8

RESULTS AND DISCUSSION

The implementation outcomes of the Secure AI Chat system confirm that artificial intelligence, lightweight cryptography, steganographic techniques, and secure communication methods can be integrated into a single, efficient solution. The testing phase showed that the emotion recognition module played a key role in the system's functionality. It operated through a layered architecture that began with a Transformer model for accurate emotion classification, utilized an NLTK-based backup for situations with limited resources, and finally used heuristic rules to maintain consistent performance across varied computational settings.

The system demonstrated strong flexibility by accurately recognizing different emotional states—such as joy, sadness, anger, fear, and neutral expressions—even when operating with limited processing resources. These detected emotions were then converted into unique XOR encryption keys, enabling a communication security method that changes based on the emotional mood of the conversation. While XOR mainly offers basic data masking, it consistently produced correct results during experimentation. Its reliability increased further when paired with Fernet, which protects saved chat records through a password-based encryption process built on AES.

This two-level encryption strategy improved overall protection while working seamlessly with the live messaging system. Additional validation was carried out through steganography experiments. By applying the Least Significant Bit (LSB) method, the system was able to conceal both text content and binary data inside PNG and BMP images without causing any noticeable change in visual quality. During testing, extraction of the hidden data was consistently flawless, indicating that the embedding technique maintained complete data accuracy and did not distort the original information.

However, Despite its strengths, evaluations also showed drawbacks when working with compressed image formats like JPG, where data recovery was affected by image compression. In contrast, the file-sharing component performed reliably, allowing smooth network transfer of different file formats and sizes without interruptions. The inclusion of a Decoy Mode added another privacy safeguard by displaying false conversation data to anyone attempting to monitor the chat, while the actual messages were securely stored in encrypted form. This feature operated effectively, indicating its value for protecting user confidentiality in security-sensitive situations.

Module-wise performance testing showed that the application consistently delivered fast message exchange, responsive interface behavior, and acceptable processing load, even during operations involving complex emotion analysis or graphical animations. In addition, the AI-driven threat detection component successfully identified messages containing abusive, harmful, or potentially risky content, thereby enhancing user protection and reinforcing overall system reliability.

8.1 Discussion

The Secure AI-Based Chat System effectively demonstrates the integration of artificial intelligence with secure communication techniques to enhance data privacy and user safety. The system successfully performs emotion detection, adaptive encryption, secure message and file transmission, and steganographic data hiding within images. Features such as encrypted chat logs and decoy mode further strengthen security by protecting sensitive information even under potential surveillance or forced disclosure scenarios. Overall, the system meets its intended objectives, offers a user-friendly interface, and provides a strong foundation for future enhancements such as improved encryption standards, higher emotion detection accuracy, and support for large-scale or multi-user communication.

CHAPTER 9

CONCLUSIONS AND FUTURE ENHANCEMENTS

The creation of the Secure AI Chat system illustrates the effective combination of artificial intelligence, encryption, and steganography to deliver a dependable, privacy-focused messaging platform. The system achieves its goals by supporting secure text communication, emotion-sensitive analysis, hidden data embedding, and safe file sharing—all within a unified and streamlined framework.

The system's multi-stage emotion recognition, combined with dual-layer encryption using emotion-based XOR keys alongside Fernet-secured logs, delivers both flexibility and strong data protection under various operating conditions. Steganography adds another layer of security by discreetly embedding information within images, while the Decoy Mode offers a creative privacy safeguard for users encountering potential monitoring or social scrutiny.

Performance testing demonstrated reliable message transmission, precise data extraction, and efficient resource management, confirming the system's suitability for real-time communication. Overall, the Secure AI Chat system stands out as a robust and innovative prototype, successfully integrating intelligence, security, and user-friendliness, and providing a solid basis for future enhancements and advanced research in secure messaging technologies.

9.1 Future Enhancement

The proposed Secure AI Chat – Emotion-Aware Communication System can be further improved by incorporating more advanced and robust security mechanisms to enhance protection against emerging cyber threats. In future versions, stronger encryption techniques such as hybrid cryptographic models combining symmetric and asymmetric encryption can be implemented to increase data confidentiality and integrity. The system can be extended beyond text-based communication by integrating secure voice and video calling features with real-time encryption.

Mobile application support for Android and iOS platforms can also be developed to increase accessibility and usability. Emotion detection accuracy can be enhanced by training customized deep learning models on larger, multilingual datasets, enabling better understanding of user emotions across diverse languages and contexts. The system may also include cloud-based encrypted storage to securely back up chat logs and allow synchronized access across multiple devices.

Additional authentication mechanisms such as multi-factor authentication or biometric verification can be introduced to further strengthen user identity verification. The application can be scaled to support group chats and multi-user communication with efficient key management techniques. Performance optimization techniques can be applied to reduce latency and improve response time, making the system suitable for real-world deployment in sensitive and high-security communication environments.

BIBLIOGRAPHY

[1] Lei Zou ., Zidong Wang ., Bo Shen ., Hongli Dong., **Secure Recursive State Estimation of Networked Systems Against Eavesdropping: A Partial-Encryption-Decryption Method,** IEEE TRANSACTIONS ON AUTOMATIC CONTROL,VOL.70,NO.6, June 2025**.**

[2] Kiran Chand Ravi., Arokia, Satish Bojjawar., V.Samuthira Pandi., Veeraiyah Thangasamy., **5G Wireless Network Security: Investigating Next-Generation Mobile Communication Data Encryption Methods and Authentication Protocols**, 3rd International Conference on Device Intelligence, Computing and Communication Technologies (DICCT) , June 11, 2025.

[3] Xinyi Shi., Yunchuan Guo., Fenghua Li., **Toward Forward- Secure End-to-End Data Sharing: An Attribute-Key-Free CP-ABE Scheme,** ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), June 11, 2025.

[4] AbelC.H.Chen., **Lattice-Based Post-Quantum Cryptography for Homomorphic Encryption,** Information & Communications Security Laboratory, Chunghwa Telecom Laboratories Taoyuan, Taiwan, 2025.

[5] Moneer M A Meftah., Yahya Al-Ashmoery., Abdul-Malik H. Y. Saad., **A Comparative Analysis of Cryptography Algorithms in Information Security,** International Conference on Computing, Engineering andDesign(ICCED)|979-8-3315-2937-6/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICCED64257.2024.10983680, June 11, 2025.

[6] Samin Salsabil., Mohammed Sadman Kabir.,Rajesh Mitra., **Secure and Decentralized Homomorphic Federated Learning for Smart Microgrid Stability,** 27th International Conference on Computer and Information Technology (ICCIT) 20-22 December 2024, Cox's Bazar, Bangladesh, December 2024.

[7] Dr.Sujata D Badiger., Akash., **Development of Data Security Algorithms in V2V Communication,** 8th International Conference on Computational System and Information Technology for Sustainable Solutions (CSITSS) | 979-8-3315-0546-2/24/$31.00
©2024 IEEE | DOI: 10.1109/CSITSS64042.2024.10816836, 2024.

[8] Liu Sheng1., Song Hangxuan1., **Design and Implementation of Energy Data Multi-party Privacy Query Scheme Based on Privacy Information Retrieval,** IEEE 4th International Conference on Information Technology, Big Data and Artificial Intelligence (ICIBA 2024).

[9] Haoling Fana,b c., Fangyu Zheng d., **Hydamc:A Hybrid Detection Approach for Misuse of Cryptographic Algorithms in Closed-Source Softwarea** State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China, 2023.

[10] Jing Zhang*., Boyu Liu., Xiaoqin Li., Xinyan Wang., Junyi Wang., Liang He., **Research on symmetric encryption and decryption algorithm of shared data based on chaotic mapping and permutation,** IEEE 6th International Conference on Information Systems and Computer Aided Education (ICISCAE),2023.

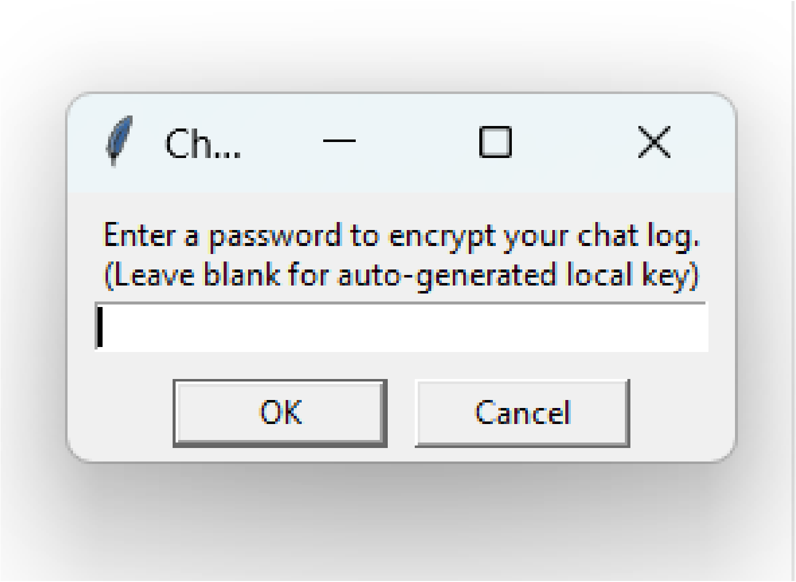
APPENDIX A

ACRONYMS

- AES : Advanced Encryption Standard
- LSB : Least Significant Bit
- AI : Artificial Intelligence
- XOR : Exclusive OR
- GUI : Graphical User Interface
- RSA : Rivest-Shamir-Adleman
- ECC : Elliptic Curve Cryptography

APPENDIX B

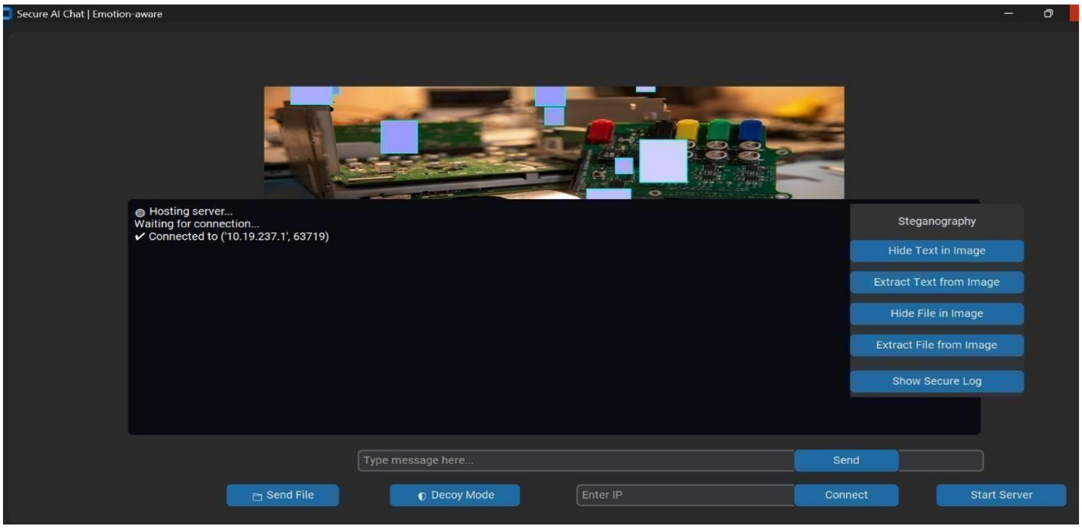
SNAPSHOTS



Enter a password to encrypt your chat log.
(Leave blank for auto-generated local key)

Fig B.1 : Entering password to chat log

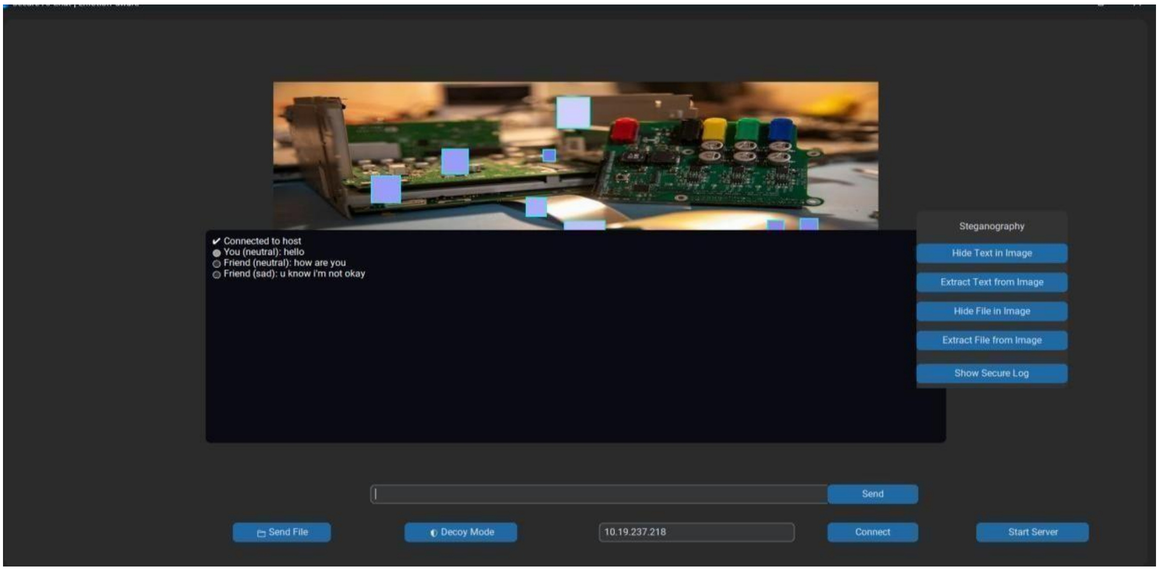
This snapshot (Fig B.1) shows a password prompt dialog box used for encrypting a chat log.



```
● Hosting server...
Waiting for connection...
✔ Connected to ('10.19.237.1', 63719)
```

Fig B.2 : Client – Server Connection

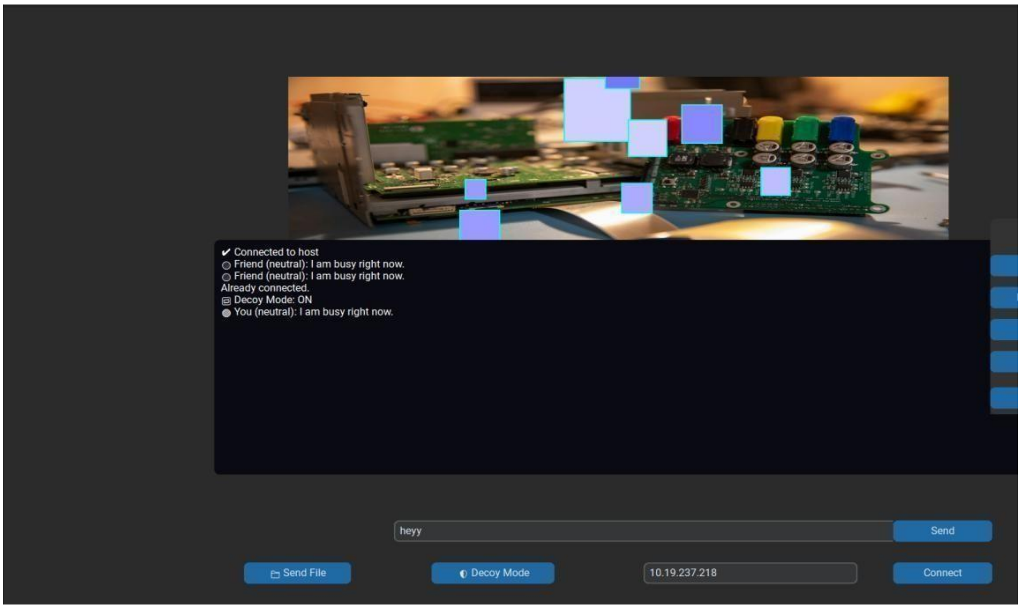
This snapshot (Fig B.2) depicts a secure client-server connection that enables encrypted message exchange, file transfer and steganography operations between connected users.



✔ Connected to host
● You (neutral): hello
○ Friend (neutral): how are you
○ Friend (sad): u know i'm not okay

Fig B.3 : Message Transfer Between Client- Server

This snapshot (Fig B.3) shows the chat box displays real-time encrypted messages between users, showing connection status and emotion-tagged conversations (neutral, sad), enabling secure and emotion-aware communication.

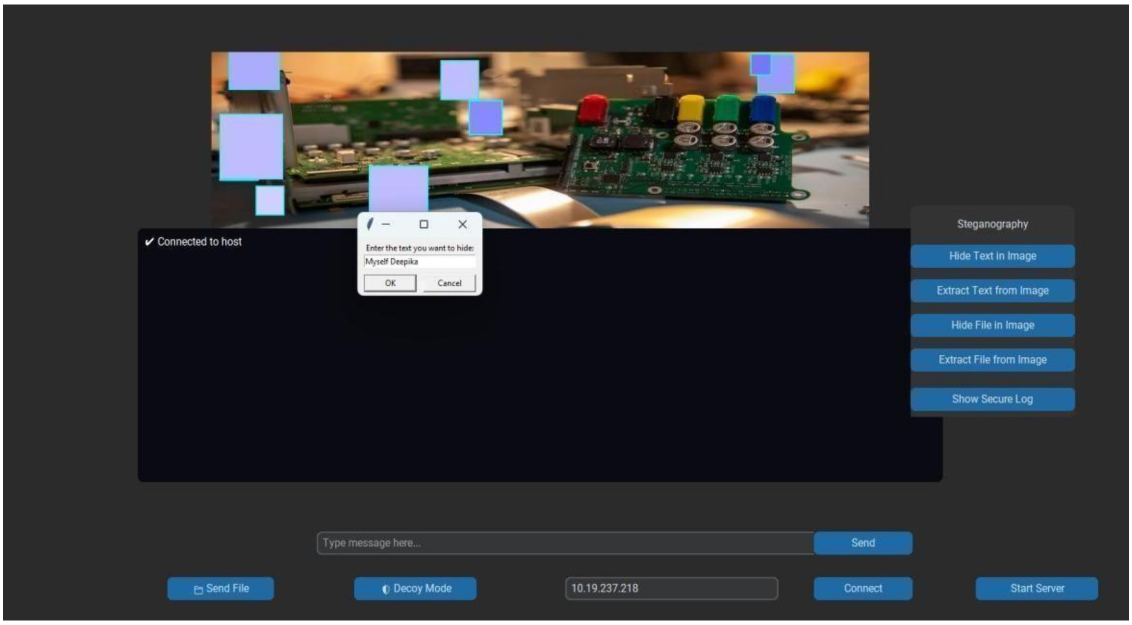


✔ Connected to host
○ Friend (neutral): I am busy right now.
○ Friend (neutral): I am busy right now.
Already connected.
Decoy Mode: ON
● You (neutral): I am busy right now.

heyy

Fig B.4 : Decoy Mode Enabled

The snapshot (Fig B.4) displays a secure chat interface showing decoy mode is enabled, Which displays fake message indicating privacy or protection features are active.



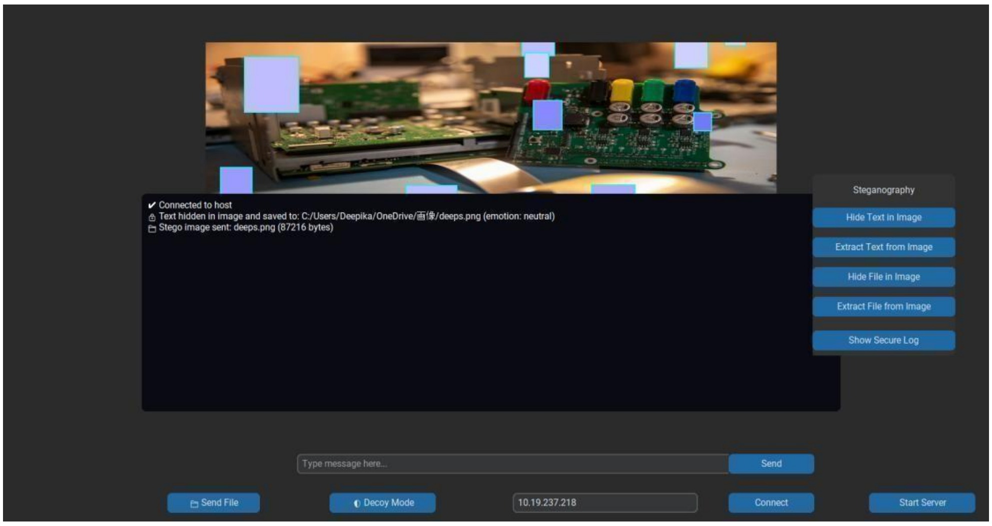
```
✔ Connected to host
```

Enter the text you want to hide:

Myself Deepika

Fig B.5 : Hide Text In Image

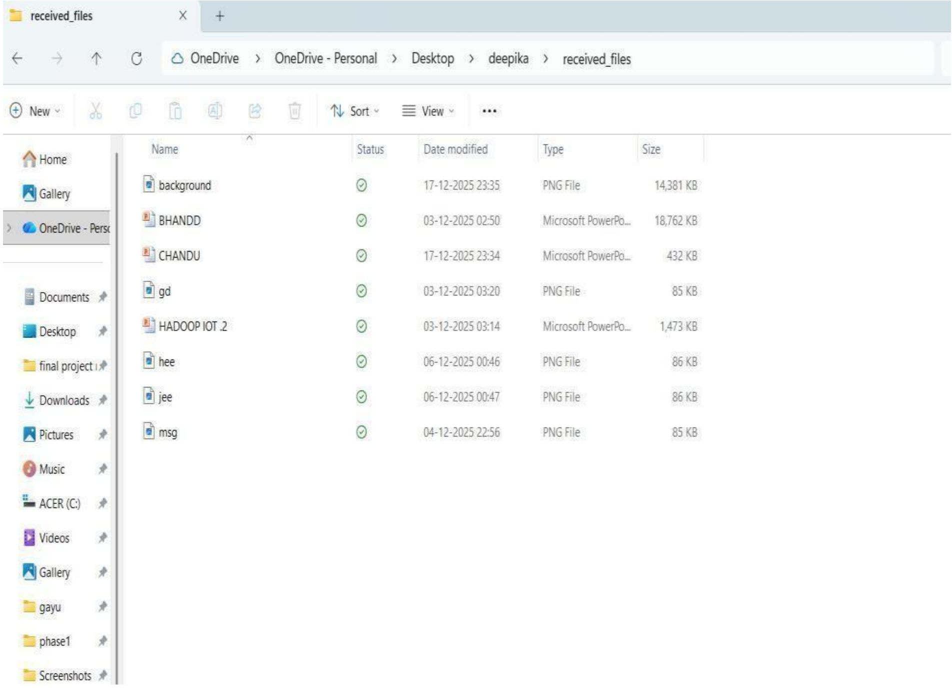
This snapshot (Fig B.5) shows hiding the text in image using steganography encryption, or file is hidden in the image.



```
✔ Connected to host
🔒 Text hidden in image and saved to: C:/Users/Deepika/OneDrive/画像/deeps.png (emotion: neutral)
🗀 Stego image sent: deeps.png (87216 bytes)
```

Fig B.6: Hide File In Image

This snapshot (Fig B.6) shows the chatbox how the data is hidden in item and send by the sender using steganography encyption.



Name	Status	Date modified	Type	Size
background	✓	17-12-2025 23:35	PNG File	14,381 KB
BHANDD	✓	03-12-2025 02:50	Microsoft PowerPo...	18,762 KB
CHANDU	✓	17-12-2025 23:34	Microsoft PowerPo...	432 KB
gd	✓	03-12-2025 03:20	PNG File	85 KB
HADOOP IOT .2	✓	03-12-2025 03:14	Microsoft PowerPo...	1,473 KB
hee	✓	06-12-2025 00:46	PNG File	86 KB
jee	✓	06-12-2025 00:47	PNG File	86 KB
msg	✓	04-12-2025 22:56	PNG File	85 KB

Fig B.7 : Received Files In Received Folder

This snapshot (Fig B.7) shows the received files folder contains various PNG images and Powerpoint files received through sender and synced with one drive indicating successfull file transfer and storage.

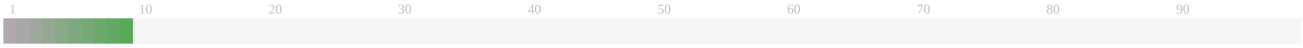
The Report is Generated by DrillBit Plagiarism Detection Software

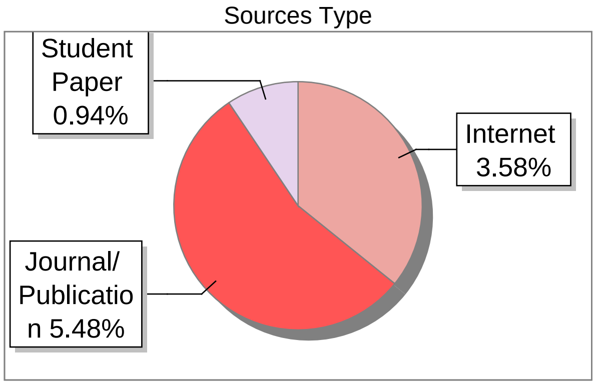
Submission Information

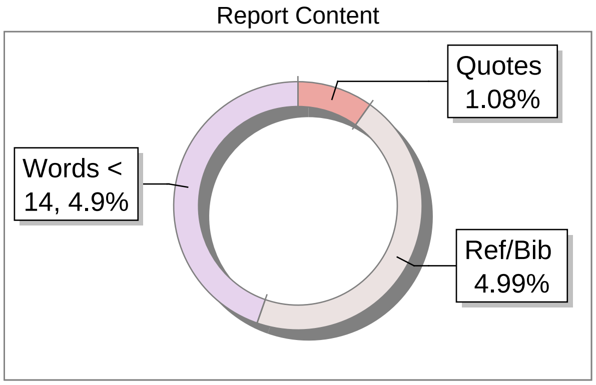
Author Name	CHANDU BHARGAVI K R, DEEPIKA U, GANASHREE J and GAYANA G
Title	DESIGN AND ANALYSIS OF CRYPTOGRAPHY COMMUNICATION SYSTEM
Paper/Submission ID	5075863
Submitted by	rameshkv.eee@gmail.com
Submission Date	2025-12-22 11:38:48
Total Pages, Total Words	62, 13086
Document type	Dissertation

Result Information

Similarity **10 %**







Exclude Information

Quotes	Not Excluded
References/Bibliography	Excluded
Source: Excluded < 14 Words	Not Excluded
Excluded Source	**0 %**
Excluded Phrases	Not Excluded

Database Selection

Language	English
Student Papers	Yes
Journals & publishers	Yes
Internet or Web	Yes
Institution Repository	Yes

A Unique QR Code use to View/Download/Share Pdf File

DrillBit

DrillBit Similarity Report

10	**82**	**A**
SIMILARITY %	MATCHED SOURCES	GRADE

A-Satisfactory (0-10%)
B-Upgrade (11-40%)
C-Poor (41-60%)
D-Unacceptable (61-100%)

LOCATION	MATCHED DOMAIN	%	SOURCE TYPE
1	sjcit.ac.in	1	Publication
2	information-science-engineering.newhorizoncollegeofengineering.in	1	Publication
3	presidencyuniversity.in	<1	Publication
4	fastercapital.com	<1	Internet Data
5	aiforsocialgood.ca	<1	Internet Data
6	Submitted to Visvesvaraya Technological University, Belagavi	<1	Student Paper
7	engg.cambridge.edu.in	<1	Publication
8	gsconlinepress.com	<1	Publication
9	arxiv.org	<1	Publication
10	Advanced firmware verification using a code simulator for the IBM System z9 by Theurich-2007	<1	Publication
12	encord.com	<1	Internet Data
13	his.diva-portal.org	<1	Publication
15	www.ijprems.com	<1	Publication
16	uifm.aulysbleu.fr	<1	Internet Data

17	www.scribd.com	<1	Internet Data
18	Submitted to Visvesvaraya Technological University, Belagavi	<1	Student Paper
19	REPOSITORY - Submitted to Visvesvaraya Technological University on 2024-07-27 21-51 2153911	<1	Student Paper
20	www.studocu.com	<1	Internet Data
21	dspace.nwu.ac.za	<1	Publication
22	lrc.acharyainstitutes.in8080	<1	Publication
23	REPOSITORY - Submitted to VEMANA INSTITUTE OF TECHNOLOGY on 2025-12-22 09-43 5074971	<1	Student Paper
25	Student Archives Data	<1	Student Paper
26	www.mdpi.com	<1	Internet Data
27	www.networkworld.com	<1	Internet Data
28	igmlnet.uohyd.ac.in	<1	Publication
29	Thesis submitted to shodhganga - shodhganga.inflibnet.ac.in	<1	Publication
30	arxiv.org	<1	Publication
31	Assessing the magnitude of soil carbon enhancement under canopy in an arboretum, by Ryan, Jaylie, Yr-2024	<1	Publication
32	docplayer.net	<1	Internet Data
33	mu.ac.in	<1	Publication
34	A photovoltaics emulator for electrochemistry using Python and SCPI, by Seidler, Martin Florian, Yr-2025	<1	Publication
35	diffy.website	<1	Internet Data

36	Thesis Submitted to Shodhganga Repository	<1	Publication
37	www.sec.gov	<1	Internet Data
38	fastercapital.com	<1	Internet Data
39	mite.ac.in	<1	Publication
40	REPOSITORY - Submitted to RAJARAJESWARI COLLEGE OF ENGINEERING on 2025-04-05 11-44 3465112	<1	Student Paper
41	springeropen.com	<1	Publication
42	www.sciencedirect.com	<1	Internet Data
43	documents.mx	<1	Internet Data
44	IMPLEMENTASI DIGITAL MARKETING PADA INSTRAGRAM SEBAGAI FAKTOR KEBERHASILAN MEDI By Trifena Siwu, Sesilia Rengkun, Yr-2025,6,21	<1	Publication
45	assets.publishing.service.gov.uk	<1	Publication
46	Distributed Learning for Heart Disease Risk Prediction Based on Key Clinical Pa By Muthu Sangeetha, Yr-2025,4,21	<1	Publication
47	Enhancement of mitochondrial carnitine and carnitine acylcarnitine translocase-, by Parvin, R, Yr-1979	<1	Publication
48	eprints.undip.ac.id	<1	Internet Data
49	fdokumen.id	<1	Internet Data
50	gisuser.com	<1	Internet Data
51	IEEE 2017 IEEE International High-Level Design Validation and Test Wo, By Barash, Guy Farchi, Eita Yr-2017	<1	Publication
52	link.springer.com	<1	Internet Data

CHAPTER 1
INTRODUCTION

In the modern digital era, communication technologies have become an integral part of personal, professional, and organizational interactions. With the rapid growth of internet-based messaging platforms, concerns related to data privacy, information leakage, cyber-attacks, and unauthorized access have increased significantly. Conventional chat applications primarily focus on message delivery and user convenience, often neglecting advanced security mechanisms that adapt to the nature of communication. As sensitive information such as confidential documents, intellectual property, and personal conversations are frequently exchanged over digital channels, there is a growing need for intelligent and secure communication systems that provide enhanced protection beyond traditional encryption methods.

This project titled “Design and analysis of cryptography communication system” aims to address these challenges by integrating artificial intelligence with secure communication techniques. The proposed system introduces an innovative approach where text-based emotion detection is used as a dynamic factor in the encryption process. By analyzing the emotional tone of user messages using AI models such as transformer-based classifiers and sentiment analysis techniques, the system adapts its encryption behavior accordingly. This emotion-driven mechanism adds an additional layer of unpredictability to the encryption process, thereby strengthening message confidentiality and making unauthorized decryption more difficult.

In addition to secure text communication, the system supports secure file transfer and steganography, enabling users to hide confidential text and files within digital images using Least Significant Bit (LSB) techniques. This feature ensures covert data transmission, making the existence of sensitive information less detectable to third parties. The application further enhances security through encrypted chat logs, ensuring that stored communication data remains protected even if local storage is compromised. To counter social engineering threats and forced access scenarios, a Decoy Mode is implemented, which displays fake messages to observers while securely preserving the original conversation in the background.

The proposed Secure AI Chat system is developed using Python, leveraging libraries such as socket programming for network communication, CustomTkinter for graphical user interface design, machine learning frameworks for emotion analysis, and cryptographic techniques for data protection.

1.1Basic Topics of the project

Projects through encrypted communication systems are based on several basic topics that form the backbone of secure data transmission. It starts with an understanding of encryption and includes the purpose, type and application in securing digital communications.

The project validates symmetric and asymmetric encryption technologies such as AES, and messages are devoured and deciphered. This includes:

Background of Secure Communication

In the modern digital era, communication has largely shifted to online platforms. Text messaging, file sharing, and image exchange are widely used for both personal and professional purposes. However, the increase in digital communication has also led to growing concerns about data privacy, unauthorized access, and cyber threats. Ensuring secure communication has become a critical requirement to protect sensitive information from interception and misuse.

Need for Intelligent and Secure Chat Systems

Traditional chat applications mainly focus on message delivery but often lack advanced security features. Many systems do not consider emotional context, behavioral threats, or hidden data protection. This creates a demand for intelligent chat systems that combine security mechanisms with artificial intelligence to enhance confidentiality, trust, and user safety during communication.

Role of Artificial Intelligence in Communication

Artificial Intelligence (AI) plays an important role in analyzing human behavior and text patterns. By applying AI techniques such as emotion detection, systems can better understand the intent and sentiment behind messages. AI-based analysis improves user interaction, enables adaptive security mechanisms, and helps detect suspicious or threatening content in real time.

Emotion-Based Message Processing

Human emotions strongly influence communication. Emotion-aware systems can identify emotional states such as happiness, sadness, anger, or fear from text messages. Integrating emotion detection into chat applications allows messages to be processed differently based on emotional context, improving personalization and enabling emotion-driven encryption techniques.

Importance of Data Confidentiality and Encryption

Encryption is a fundamental method used to secure digital data. By converting readable data into an unreadable format, encryption ensures that only authorized users can access the information.